

Adobe Commerce Implementation Playbook

Created by: Scott Rigby



In partnership with:
dentsu



Table of contents

ADOBE COMMERCE IMPLEMENTATION PLAYBOOK

01 Introduction	3
02 Project Scoping	12
03 Development and Quality Control Procedures	28
04 Planning and Governance	45
05 Architecture and Integrations	58
06 Infrastructure and Deployment	92
07 Launch and Cut-Over Processes	130
08 Ongoing Support and Maintenance	138



Introduction

ADOBE COMMERCE IMPLEMENTATION PLAYBOOK

01



Your end-to-end guide to Adobe Commerce

The aim of this playbook is to provide the most holistic overview of a typical Adobe Commerce implementation.

From the early stages of project scoping through the development and integrations and finally to deployment and ongoing support, there are many methodologies and best practices that need to be considered to successfully launch a commerce project.

Additionally, these processes and considerations apply to any kind of Adobe Commerce project:

- Small, medium, or large implementations
- B2C, B2B, and B2B2C business models
- Monolithic or headless architectures
- Single-market or multi-market rollouts
- Extensive integrations with or without middleware

We hope that this playbook can provide insights and guidance to a variety of stakeholders who would typically be involved in an ecommerce project initiative, such as:

- CEOs and general management, who should have a solid idea of what an ecommerce rollout entails
- CMOs and digital managers, who will be working with business users on the platform itself
- CTOs and technical managers, who will be heavily involved during all stages of a technology project implementation
- Project managers and project champions who lead a commerce project initiative and should have all relevant information at hand

While the success of an IT project depends very much on the experience and expertise of the team (typically a solution partner) that develops, customizes, integrates, and maintains the code, we believe it is relevant for all stakeholders to familiarize themselves with the best practices of an Adobe Commerce implementation. We hope you enjoy this read.



Sections of this playbook

The structure of this playbook follows the typical lifecycle of an Adobe Commerce implementation project. This simplifies navigation throughout this document, as readers can immediately skip to the relevant section of a project for all relevant information.

Project Scoping

A breakdown of the key stakeholders, processes, timelines, and requirements that are critical for brands to understand and complete for a successful implementation.

Development and Quality Control

A look at the tools, solutions, processes, and methodologies that have been tested and perfected over a number of Adobe Commerce implementations, as well as our recommendations for which solutions are best to fit specific business needs and objectives.

Planning and Governance

A dive into creating a plan for delivering a solution on time, on budget, and that meets your needs is instrumental to success.

Architecture and Integrations

The capabilities, architecture, and integrations that make Adobe Commerce one of the most trusted and reliable ecommerce platforms on the market.

Infrastructure and Deployment

Moving further into the actual platform itself, we highlight the infrastructures and environments that are powering Adobe Commerce and the software solutions that make it such a robust platform.

Launch and Cut-Over Process

The tactics and actions from pre-launch to post-launch that must take place to ensure the site goes live and maintains its level of effectiveness from day one onward.

Ongoing Support and Maintenance

Details around the transition phase and the types of models and SLAs around an ongoing support plan to continue moving your brand forward after the launch.



About Adobe Commerce

Adobe Commerce gives merchants the power to create unique and engaging shopping experiences. The platform provides rich, out-of-the-box features, an unlimited ability to customize, a flexible headless architecture, and seamless third-party integrations. This is coupled with a global ecosystem of implementation partners and a vast marketplace of extensions to bring commerce projects to life.

Delivering meaningful commerce experiences

Marrying content with commerce to meet customer demands for flawless brand interactions, Adobe Commerce's features are always evolving with the consumer in mind. Adobe Commerce allows brands to own your customer experience, craft personalized content and promotions, and deliver a smooth path to purchase.

- Page Builder
- Product recommendations
- Customer segmentation & personalization
- Content staging & preview
- Instant purchase
- Merchandising

Conducting seamless shopping everywhere

Commerce growth is fueled by serving your customers' needs wherever they are. Expand your reach and grow into new channels, geographies, and sell online to both businesses and consumers in one platform. Bring commerce to your customers wherever, whenever, and however they prefer to shop.

- Mobile commerce
- B2B ecommerce
- PWA Studio
- Global expansion
- Amazon sales channel
- ERP integration
- Headless commerce architecture

Managing intelligent and efficient operations

Efficiency, automation, and access to data are required to compete in the modern commerce landscape. Adobe Commerce makes it easy for our merchants to transform into, and operate as, an effective, data-driven commerce organization to grow their business.

- Business intelligence
- Shipping
- Inventory management
- Order management and customer service

Operating with confidence

Shoppers have high expectations for the brands that they engage with—from high performance and security to innovative interactions. Run your commerce operations with the confidence that you will be able to meet and exceed those expectations.

- Open & customizable
- Cloud deployment
- Security & trust
- Performance & scale
- APIs
- Marketplace

Adobe Commerce: Principals of Platform Development

Like any large IT project, Adobe Commerce is built on coding standards that leverage best practices and standardizations of the underlying technologies (PHP/Zend, symfony, JavaScript, jQuery, HTML, etc.), as well as standards that have been established within the Adobe Commerce Coding Standard. Following these standards is an absolute must to eliminate bugs and improve the quality and maintainability of the custom-built code.

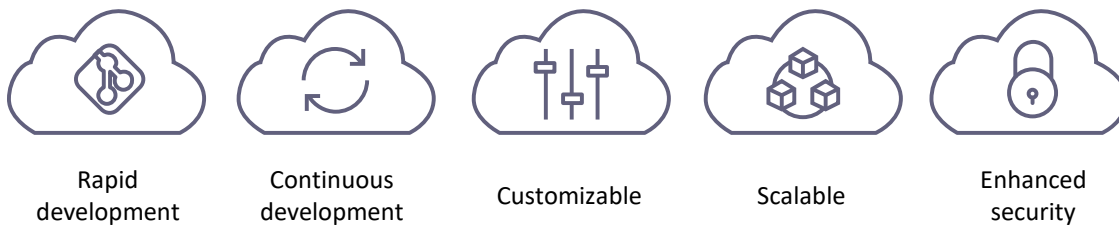
Within this playbook, we are diving deeper into some of the main standards of Adobe Commerce development. This includes:

- Functional and technical scoping in line with the development process
- Development best practices aligning with the MVC architecture
- Architectural considerations, including GRA
- Security standards against scripting, exploits, etc.
- Extension development best practices
- Web API integrations with REST, SOAP, GraphQL
- Performance improvements for coding and infrastructure
- Testing tools, strategies, and methodologies

While some solution implementers may have their own preferences when it comes to the methodologies, processes, and tools used throughout an implementation project, we are focusing this playbook on generally accepted best practices & methodologies that could be shared across the majority of implementations.

Adobe Commerce Cloud

Adobe Commerce Cloud is a managed, automated hosting platform for the Adobe Commerce software. Adobe Commerce Cloud comes with a variety of additional features that sets it apart from the on-premise Adobe Commerce and Magento Open Source platforms:



Adobe Commerce Cloud provides a pre-provisioned infrastructure that includes PHP, MySQL, Redis, RabbitMQ, and Elasticsearch technologies; a Git-based workflow with automatic build and deploy for efficient rapid development and continuous deployment every time code changes are pushed in a Platform as a Service (PaaS) environment; highly customizable environment-configuration files and tools; and AWS hosting that offers a scalable and secure environment for online sales and retailing.



Project Scoping

ADOBE COMMERCE IMPLEMENTATION PLAYBOOK

02

Knowledge is power

The commerce experience has expanded to include more devices and channels than ever before, while advancements in technology have made the journey from awareness to purchase as fast as a couple of clicks. Brands must transform their ecommerce approach to be everywhere their customers are as part of a seamless, cohesive customer experience. This requires more than just a facelift to your ecommerce site.

Achieving this transformation is not simple, nor is it cheap. It requires an investment of time, budget, and resources, as well as complete buy-in from the top to the bottom of the organization. In order to get that buy-in, brands must understand what it needs and how to reach its goals. The project-scoping phase is an integral part of the ecommerce journey that takes place well before a single metaphorical wrench is turned but can make or break the entire process.





Key stakeholders

Many businesses in 2021 have come to realize that becoming “pandemic-proof” directly relates to an organization’s digital strength. Based on Forrester data, [McKinsey](#) found that ecommerce has seen 10 years' growth in as little as 3 months, with growth being driven by the need to connect digitally with customers while brick-and-mortar doors were locked shut.

When the decision is made to invest in a commerce transformation, one of the most important steps is determining who from the organization is involved in the process. For smaller businesses, sometimes the ecommerce or digital team is only a handful of people, so the entire team takes on the task. However, for larger organizations, one approach is to split this up into three different areas.

Who owns the experience?

This is the person or team at the organization with the best understanding of the current and future wants and needs of the customer and who is ultimately responsible for delivering the experience that addresses those wants and needs. This includes focal points like product assortment, inventory visibility, promotions, customer complaints, and real-time order updates and shipment tracking. This group will often carry titles such as:

- VP of Digital Strategy
- VP of Ecommerce
- VP of Marketing
- Customer Service Manager
- Head of Sales (mainly for B2B applications)





Who owns the technology?

Most commerce implementations will require several different platforms to communicate back and forth and be fully integrated, such as an order management system, customer service platform, or ERP system. Back-office systems being prepared for a commerce transformation will often dictate the timeline for an implementation, so having contacts from an organization's IT team involved from day one to support the initiative is imperative. This group will often include:

- IT Systems Manager, CTO, and in-house tech team
- ERP (or any back-office) Engineer

Who is the executive sponsor?

Digital Commerce 360 has reported that the No.-1 reason that organizations haven't yet invested in a new commerce transformation was because it lacked leadership support. The rate that the leadership team is involved is crucial to the brand's success. Full support from the following roles, including recurring checkpoint meetings throughout the duration of a commerce implementation, is recommended:

- The entire C-suite



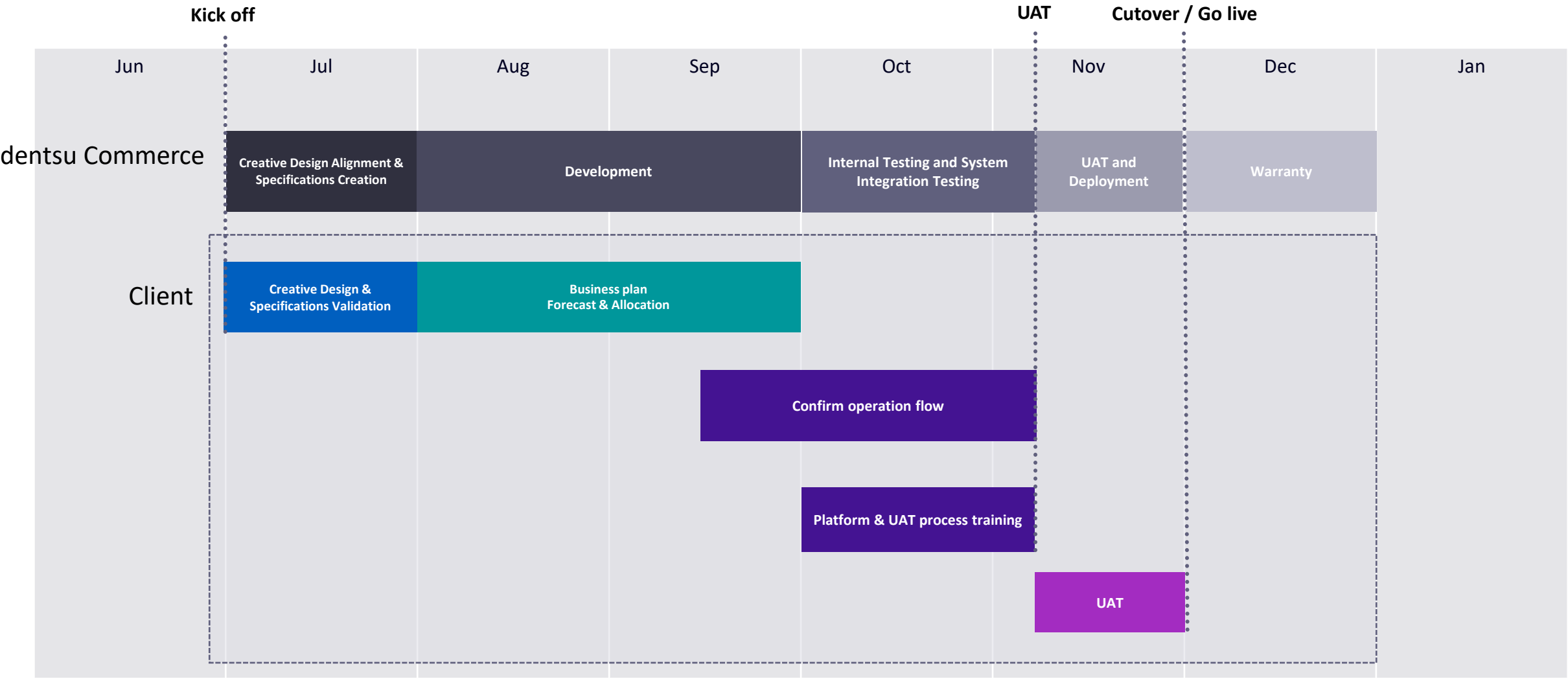
Typical process and timeline

Many organizations prepare for a commerce transformation by creating a request for proposal. This will include detailed requirements, current systems of record, a business case, and specific goals (revenue, conversion, average order value, etc.) to define success, and then the brand will approach 2-3 different commerce systems integrators or digital agencies to evaluate, review estimates and proposals, select a company, and begin working toward a projected timeline to launch.

This can be a successful approach. However, one recommendation that mitigates risk (before committing to a full implementation) is to work with an organization on a requirements assessment or discovery process that can confirm organizational preparedness, project scope, timeline, budget, business requirements, and technical approach. On average, this process should take between 4 and 6 weeks, but will vary depending on the size of the project.



Typical timeline example



Sample timeline activities

Discovery/ Define/ Design	Development	Internal Testing & System Integration Testing	UAT/ Warranty
Key Activities			
<ul style="list-style-type: none"> • Discuss high level scope • Workshop to understand requirements for design, customer features and third-party integration • Clarify requirements and suggest solutions • Define final scope • Support the design activities by adapting design to platform standards • Create specifications • Clarify hosting provider 	<ul style="list-style-type: none"> • Weekly follow-up and active management of stakeholders, partners and internal team starts from here until launch • Develop the solution • Develop the testing strategy • Record CRs as necessary and discuss with customer how to handle (after go-live, within the release but extent the timeline) 	<ul style="list-style-type: none"> • Conduct internal quality control testing • Execute SIT plan • Log and track defects until closed • Perform regression testing 	<p>UAT</p> <ul style="list-style-type: none"> • Support UAT team • Triage UAT defects (showstoppers, non-blocking) • Final regression <p>Warranty</p> <ul style="list-style-type: none"> • Monitor site after go-live • Support customer with existing defects • Run sanity after delivering new code
<p>Deliverables</p> <ul style="list-style-type: none"> • Business requirement document • Functional specifications • Technical architecture between all systems • Technical specifications • Detailed project plan for development 	<p>Deliverables</p> <ul style="list-style-type: none"> • Code packages • Weekly status reports • Functional test cases/scripts • E2E SIT test cases/scripts • Hosting ready (UAT and Live Servers) 	<p>Deliverables</p> <ul style="list-style-type: none"> • SIT processing • Know issues non-UAT blocking 	<p>Deliverables</p> <ul style="list-style-type: none"> • UAT demo/training • UAT processing • PUA (Project User Acceptance)

Deliverables after completing a discovery process should include the following:

1. **Business Requirements Document:**
This document captures and obtains business requirement approvals around functional specifications and the usability of the consumer experience. It is used to drive the user experience, visual design, and detailed testing plans.
2. **System Diagram:**
A systems diagram should show data (orders, products, third-party integrations) flowing to and from the various systems involved (Adobe Commerce, OMS, ESP, payment providers, etc.).
3. **Technical Specification Document:**
This document will define a technical-implementation approach and applicable details about all third-party integrations and systems that will need to be modified. This includes a set of APIs that don't currently exist and will need to be developed.
4. **Data Mapping:**
A data mapping document covers categories, products, recommendations, pricing data, and Inventory, and includes the necessary attributes to map the current products and categories to Adobe's product catalog.
5. **Flow Diagram:**
Showing user flows for critical and/or unique processes especially when customized from standard Adobe Commerce behavior
6. **Finalized Scope and Contracted Third-Party Vendors:**
This is a finalized list of named third-party vendors to be integrated into the Adobe Commerce solution.
7. **Product Data:**
A review of an organization's existing product, pricing, and inventory data, as well as a plan to map data attributes used to populate Adobe Commerce.
8. **Project Plan:**
A project plan will detail the high-level tasks and timelines associated with the implementation initiative, including overlapping integrations and third-party dependencies. This plan may also include a responsibilities matrix, or "RACI," chart that shows who involved is responsible, accountable, consulted, and informed.

Requirement Checklists

The following questions can serve as a starting point to see which information should be documented to confirm organizational preparedness. They can also help when developing an RFP:

Business Questions:

- What are the business goals for the new ecommerce platform?
- What are the reasons for changing your current ecommerce platform?
- What are the objectives for the website infrastructure?
- What is your timeframe to deliver the new ecommerce platform?
- How many of your IT project managers will be assigned to this project?
- How many of your business analysts will be assigned to this project?
- How many of your technical analysts will be assigned to this project?
- How many of your HTML developers will be assigned to this project?
- What documentation exists for current business processes?
- What documentation exists for current system integrations?
- What documentation exists for current process flows?
- What documentation exists for current data flows?
- Who will provide the training?
- Which training will be completed prior to go-live?
- Which training will be completed after to go-live?
- What Adobe Commerce support will be required after go-live?
- Is this project dependent on other system development projects?
- How do you see your sales growing over the next 12-24 months?
- Who are your main competitors? Please provide links to their sites. How do you want to differentiate your online experience from your competitors?
- Where do you see future growth coming from in your business?
- What role does digital commerce play in your business strategy? What are your primary objectives for setting up this ecommerce platform?
- Do you have any brands/companies that you take as a reference on how you grow your omnichannel business?
- Which teams or individuals are driving the ecommerce strategy? Describe the relevant positions.

Current Platform:

- How is the current platform being hosted: Internal, hosting provider, private cloud servers, or hosted cloud servers?
- Which environments does the current platform have: development, QA, pre-production, production?
- How many web and database servers are in the development environment?
- How many web and database servers are in the QA environment?
- How many web and database servers are in the staging (pre-production) environment?
- How many web and database servers are in the production environment?
- Is it a multi-server architecture with load balancing?
- Is it a high-availability system architecture?
- What problems are you experiencing with your current websites?
- Current catalog size (number of SKUs)?
- Average visitors per day?
- Average concurrent sessions per hour?
- Average page views per day?
- Average orders per hour and per day?

Expected Platform Requirements:

- Which Adobe Commerce version will you use?
- How will the future platform be hosted: Internal, hosting provider, private cloud servers, or hosted cloud servers?
- Which environments will the future platform have: development, QA, pre-production, production?
- How many web and database servers will be in the development environment?
- Will it be a high-availability system architecture?
- How many Adobe Commerce administrators will you have?
- Expected catalog size (number of SKUs)?
- Expected average visitors per day?
- Expected average concurrent sessions per hour?
- Expected average page views per day?
- What data should be imported from the old site to the new site? (Example: product, customer, order history)

Websites:

- How many domestic websites will be implemented?
- Which languages will be implemented?
- How many international websites will be implemented?
- Which regions will the websites support?
- Which languages will be implemented?
- Which currencies will be implemented?
- Do you have delivery service-level agreements for each country?
- Who are your delivery providers for each country?
- Who are your tax-accounting providers for each country?
- Will you require a custom international website theme?
- Is this primarily a B2C or B2B site? Is there any B2B2B or B2B2C element?
- Is there an existing design that is adapted, or will the platform be designed from scratch?
- Is there a requirement for headless commerce (separate frontend and backend layers)?
- What are the break points (tablet/mobile) for the responsive design?
- Is there a requirement for a mobile app? Should PWA be leveraged for mobile frontend?
- Any specific browsers that should be tested (except for the standard browsers IE9+, Firefox, Chrome, Safari)?
- What are the language(s) for each front end? Is the translated content available or is support needed?
- Are there multiple websites? If so, can customers use their credentials across all sites?
- Is product data shared across all sites?
- Are there changes in the design from site to site?
- Are there subscription options and how are subscribers managed?
- Are there specific SEO requirements? How is search engine marketing managed?

Integration:

- Which CMS system will be integrated with Adobe Commerce? (Examples: WordPress, Drupal, Concrete5)
- Are there existing APIs that can be used?
- Has system-error handling been designed and developed for this third-party-system integration?
- Which ERP system will be integrated with Adobe Commerce? (Examples: SAP, MS Dynamics NAV)
- Which shipping carrier system will be integrated with Adobe Commerce?
- Which tax-software system will be integrated with Adobe Commerce? (Example: Taxware)
- From which system will product data be imported into Adobe Commerce?
- Frequency of imported product data loads?
- Into which system will Adobe Commerce export product data?
- Frequency of exported product data loads?
- From which system will order data be imported into Adobe Commerce?
- Frequency of imported order data loads?
- Into which system will Adobe Commerce export order data?
- Frequency of exported order data loads?
- Which analytics platform is used? (Examples: Adobe Analytics, Google Analytics)
- Will there be single sign on and social network sharing? If so, which social networks?
- Is there any rewards or loyalty programs in place? Is there any integration with POS systems for rewards?
- How are product returns handled? Is customer expected to log a return request online?
- Is an online chat tool needed? Is a chatbot needed?
- Which payment gateway do you want your site to have?
- Which order management system will be integrated with Adobe Commerce? (Examples: Microsoft Dynamics, SAP, Retail Pro)
- Which product inventory management system will be integrated with Adobe Commerce? (Examples: Akeneo, InRiver, Bluestone)
- Which customer relationship management system will be integrated with Adobe Commerce? (Examples: Hubspot, Salesforce, Klaviyo)

Adobe Commerce-Specific Features:

- Will you require any sort of A/B testing?
- How many concurrent administrators can be working in the system?
- Will you allow a customer to pick up items purchased on the website at a store?
- Will you have promotional pages?
- Will you have marketing banners?
- Will you want to have videos play in a CMS page?
- What will be the frequency of modifying or updating content?
- What type of content will be loaded?
- Will content updates be scheduled?
- Will you require a content staging website?
- Should customers be allowed to create a website account
- Will you use unique discount coupons for promotions?
- Will you have promotional pricing?
- Will you have flexible coupons (ability to set per website, customer group, time, categories, or products)?
- Will percentage discounts be offered for single items?
- What type of gift functionality will be required?
- Will rewards-program functionality be required?
- Will newsletter functionality be required?
- Allow a customer to view past orders and select items to be exchanged?
- Will you allow a customer to initiate the cancelling of an order from the website?
- Will you allow a customer to initiate the exchange of items from the website?
- Will you require real-time address validation?
- Will you allow a customer to initiate the return of items from the website?
- Will Adobe Commerce issue a return RMA?
- Capture refund information in Adobe Commerce?
- Will you require online order tracking for a registered customer?
- Will you require online order tracking for a guest customer?
- Execute online authorization and capture during order placement?
- Authorization with fraud filter during order placement with delayed capture
- Number of days to receive full payment (not authorization)?
- Capture and execute against expiration of authorization?
- Authorize.net
- Braintree

Adobe Commerce-Specific Features: (cont.)

- PayPal Express
- Store credit
- Plastic gift cards
- Reward points
- "Bill Me Later" – more commonly known as "Buy Now, Pay Later" as it's immediately billed but not paid yet
- Will there be different product pricing on different websites?
- Will functionality to compare different products be required?
- What types of products will you sell?
- What will be the frequency of adding new products?
- What will be the frequency of updating products?
- What will be the frequency of creating new categories or sub-categories?
- Will "Ratings and Reviews" functionality be required?
- Will you allow customers to recommend a product?
- Sales: Orders, tax, invoiced, shipping, refunds, coupons, PayPal settlement reports
- Shopping cart: Products in carts, abandoned carts
- Products: Bestsellers, products ordered, most viewed, low stock, downloads
- Customers: New accounts, customers by orders total, customers by number of orders, customer segments, customer reviews
- Products reviews
- Tags: Customers, products, popular
- Search terms
- Invitations: General, customers, order conversion rate
- Will you need Adobe Commerce to generate reports based on coupon usage data?
- Will you need Adobe Commerce to generate reports based on sales data?
- Will you need custom Adobe Commerce reports?
- What is your current SEO strategy?
- What will be your new SEO strategy?
- What are your requirements for SEO migration?
- Store fixed rates in Adobe Commerce?
- Allow partial shipping?
- Should shipment tracking information be stored in Adobe Commerce?
- Will you require tax-calculation rules for your domestic regions?
- Will you require tax-calculation rules for your international regions?



Development + Quality Control Procedures

ADOBE COMMERCE IMPLEMENTATION PLAYBOOK

03

Ensuring Excellence

There is no shortage of aspects that must be well thought through and rigorously tested to keep an ecommerce site running without interference. Not only must you identify the right solutions to tackle every aspect of the site—from data storage and programming to caching and security—but you need the right process to ensure the delivery of a platform that both runs smoothly and can be built and optimized efficiently.

This section will offer not only a look at the tools, solutions, processes, and methodologies that have been tested and perfected over a number of Adobe Commerce implementations, but also our recommendations of which solutions best fit specific business needs and objectives.



Purpose	Tool
Database	MySQL, MariaDB, Percona
Programming Language	PHP, JS, HTML, LESS CSS
Integrated Development Environment	Eclipse and PHPStorm
Web server	Nginx, Apache
Caching services	Redis, Varnish
Search Services	Elasticsearch
Message Queue Services	RabbitMQ
Security tool scan	SonarQube, ZAP

Tools and Platforms

First, let's go through our recommended solutions that can be used within Adobe Commerce to drive performance on the platform.

Database

There are three different tools that we use depending on the needs of the brand. MySQL is a great baseline solution as the Adobe Commerce database if you don't expect your store to handle extreme loads. Meanwhile, MariaDB is more community-focused and works better for users who care more about features than pure performance. MariaDB supports a large array of database engines, disk encryption, complex horizontal interconnectivity, and scaling features, which could be interesting for large Adobe Commerce stores. Finally, Percona is a fork of MySQL that centers around performance and peak load handling. Choose MariaDB if you need more quality of life and DevOps features. Go for Percona if your goal is to gain high-load performance in large-scale datasets.

Programming Language

Adobe Commerce core is a PHP-based application, and the newest releases are always compatible with the latest stable PHP version (e.g., Magento 2.4 recommends using PHP7.4). To get more security and performance, there are several factors to account for when configuring PHP to get maximum speed and efficiency on requests processing. Adobe Commerce web storefront is built with HTML, JavaScript, and LESS CSS pre-processor.

Web Server

Adobe Commerce fully supports the Nginx and Apache web servers. Adobe Commerce provides sample recommended configuration files in the `<magento_home>/nginx.conf.sample` (Nginx) and `<magento_home>.htaccess.sample` (Apache) files. The Nginx sample contains settings for better performance and is designed so that little reconfiguration is required.

Caching Services

Adobe Commerce provides numerous options to store your cache and session data, including Redis, Memcache, filesystem, and database. For a setup with multiple web nodes, Redis is the best option.

Adobe Commerce highly recommends using Varnish as the full-page cache server for your store. Adobe Commerce distributes a sample configuration file for Varnish that contains all recommended settings for performance.

Message Queue Services

Message queues provide an asynchronous communication mechanism in which the sender and the receiver of a message do not contact each other. RabbitMQ is an open-source message broker that offers a reliable, highly available, scalable, and portable messaging system.

Search Services

From Magento 2.4 on, all installations must be configured to use Elasticsearch as the catalog search solution. Elasticsearch provides quick and advanced searches on products in the catalog. Before Magento 2.4, it's also recommended but is optional.

Security Tool Scan

The Adobe Commerce Security Scan Tool enables you to regularly monitor your store websites and receive updates for known security risks, malware, and out-of-date software. Typically, you start using this tool when you begin user-acceptance testing (UAT). Besides the Adobe Commerce Security Scan service, which is a free service available for all implementations and versions of Adobe Commerce Cloud, there are other choices that can be used during the CI/CD process and before each release.

SonarQube is an open-source quality management platform, designed to analyze and measure your code's technical quality. SonarQube not only provides a complete report of code bugs, syntax errors, and vulnerabilities, but also offers suggestions and examples about how to fix your code. SonarQube is perfect to use in a CD/CI environment as a tool capable of analyzing the code before it's deployed.

Zed Attack Proxy (ZAP) is a free security testing tool used by thousands of pen-testers around the globe. ZAP is developed by OWASP and is one of the most preferred tools for manual security testing.

Purpose	Tool
Ticket Management	Atlassian JIRA
SLA Management	Atlassian JIRA Service Management
Document Management	Atlassian Confluence
Source Code Management	Atlassian Bitbucket, GitHub
Continuous Integration (CI) / Continuous Delivery (CD)	Jenkins
Communication	Email, Teams, Phone, Slack
Meeting	Zoom, Teams

Ticketing/Documentation/Source Code Management

Moving out of the actual Adobe Commerce environment, but equally important, are the tools and solutions that make collaboration and the entire development process as smooth as possible.

SLA Management

Jira Service Management empowers development and operations teams to collaborate at a high velocity so they can respond to business changes and deliver great customer and employee service experiences fast.

Document Management

Confluence is great tool to build, organize, and collaborate on work in one place from virtually anywhere. Confluence comes equipped with best-practice templates, so you don't need to reinvent the wheel. Get started easily and maintain consistency across teams and projects.

Ticket Management

Jira is a ticket tracking and management system (including version and release management) with powerful features, such as:

- Constraint workflow to avoid mistakes
- Estimation and time-sheet tracking at ticket level
- Easy ticket visualization and moving with Agile/Scrum
- Transparent information among client team and development team

Source Code Management

Bitbucket is more than just Git code management. Bitbucket gives teams one place to plan projects, collaborate on code, test, and deploy. Using Bitbucket, you can approve code reviews more efficiently with pull requests. You can also create a merged checklist with designated approvers and hold discussions right in the source code with inline comments. GitHub is similar to Bitbucket, which is also used in some projects.

Communication

For different clients, we may use a variety of tools for communication: email, Microsoft Teams, and Slack are all common tools that we have leveraged for project-related communications.

Meeting

Microsoft Teams and Zoom are the most common tools to set up online meetings.

Continuous Integration (CI) and Continuous Delivery (CD)

Jenkins is the leading open-source automation solution, Jenkins provides hundreds of plugins to support building, deploying, and automating any project. Jenkins can be used as a simple CI server or turned into the continuous delivery hub for any project.



Project implementation methodology

Now that you have a better idea of the tools that are involved, we will now break down our delivery and testing processes.

Continuous Integration/Continuous Delivery Process

Continuous Integration (CI):

- A job will be created automatically to perform the following actions:
 - Building the source code to check compilation errors.
 - Executing tests automatically when a pull request is created/updated. At present, the PHP unit test will be executed.

The job execution's status will be updated to the pull request. Developers can view the detail of the job execution so that they can fix/improve existing coding.

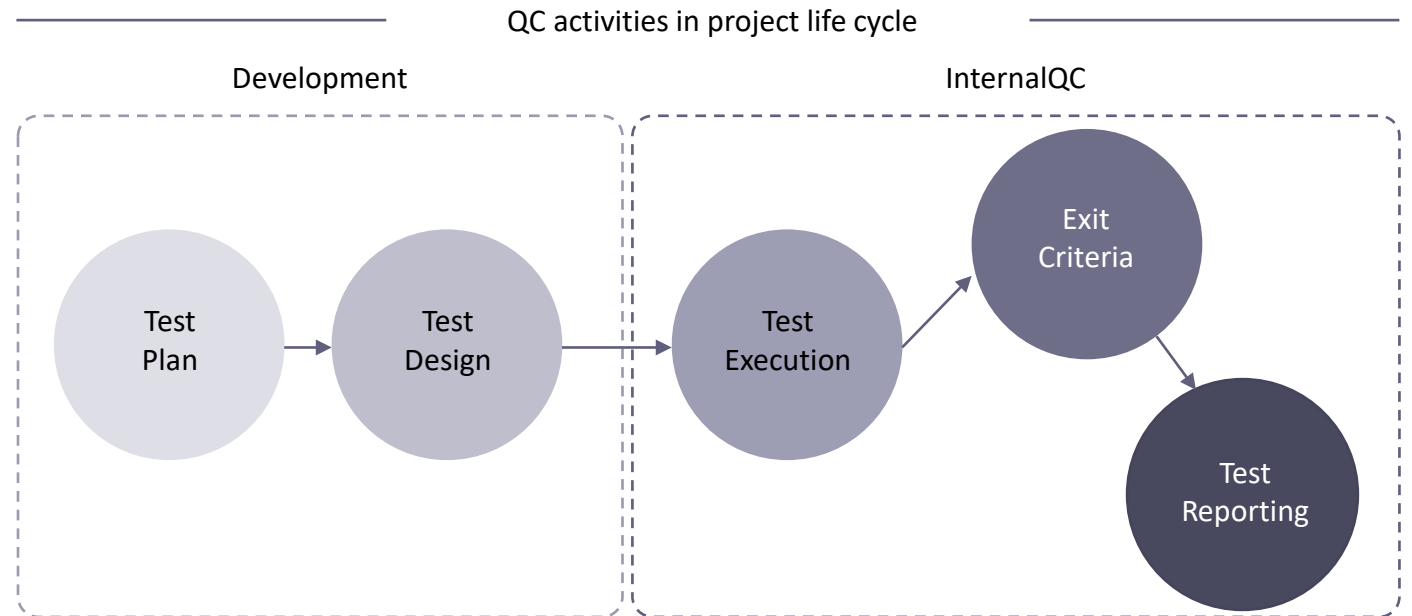
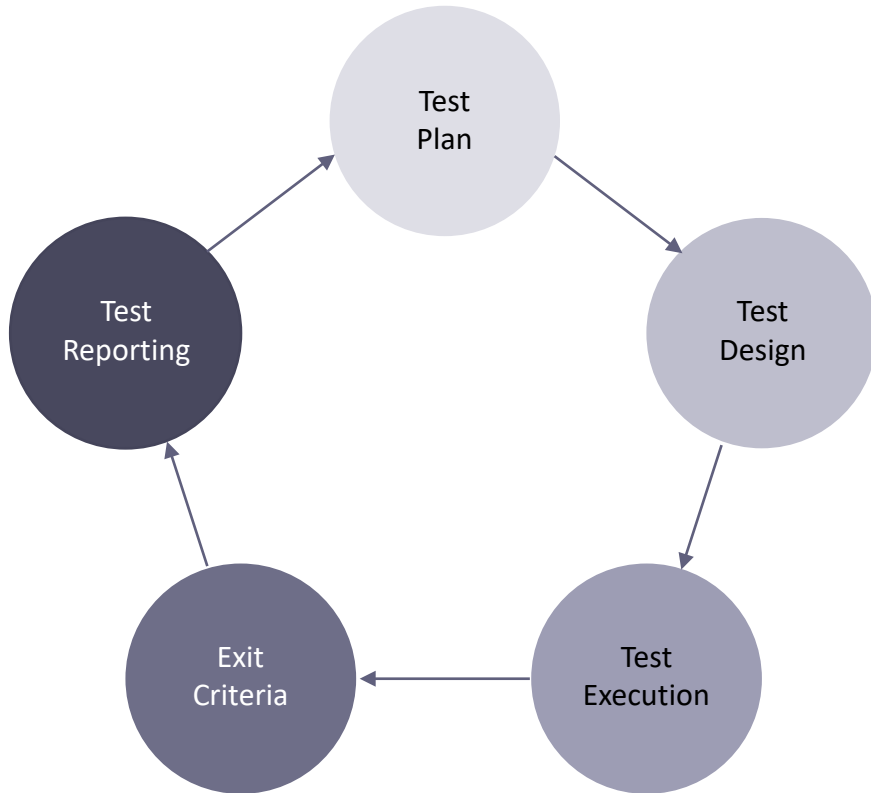
As the build was executed on the build system, it will not only minimize the deployment downtime, but also reduces the load of the server. As a result, the QA activities, which are happening on the server, will be less impacted.



The CI/CD process in the previous diagram can be briefly described as follows:

- Bitbucket acts as the Git repository.
- Docker images will be replicated from production technology stack configurations.
- Docker containers will be used for all development, testing environments. Other environments could leverage these configurations if needed.
- Developers will perform a checkout of the relevant code branch for every new task/ticket.
- For all commit branches, we will automatically:
 - Execute a standard code scan
 - Execute a code compiling test
 - Executed static code scan (e.g., SonarQube)
- All PASSED scan commits will be merged to the target branch.
- A new released tag will be pushed to S3 for deployment readiness package.
- New deployment will be triggered by our deployment engineering team.
 - A deployment job will deploy the new package to the target environment.
 - Database structure updates will require a pause to take on new requests from the customer.
- The deployment process ends with an email/Slack/Teams notification, sent automatically by the server to the deployment engineering team.

Quality Control Process



The quality control process in the previous diagram can be briefly described as follows.

Software Development Process	QC Workflow	QC	QC Leader
Development	Planning		<ul style="list-style-type: none"> Review and contribute to test plans Create test specifications (test cases/test scenarios) Prepare and acquire test data
	Test Analysis and Design	<ul style="list-style-type: none"> Review and contribute to test plans Create test specifications (test cases/test scenarios) Prepare and acquire test data 	<ul style="list-style-type: none"> Initiate the preparation, specifications Write or review a Test strategy for the project Leading, guiding and monitoring the analysis, design
Internal Testing	Test Implementation and Execution	<ul style="list-style-type: none"> Implements tests, execute and log the tests^[1]_{SEP} Check performance and scan security^[1]_{SEP} Evaluate the results and the deviations from expected results^[1]_{SEP} Post bugs to bug tracking system (Jira/Redmine/Trello)^[1]_{SEP} Re-testing (confirmation testing) after bug fixing 	<ul style="list-style-type: none"> Monitoring implementation and execution of the tests Ensure traceability of the tests to the test basis and keep track of bugs on the Bug tracking system Prioritize/schedule tests to align with the project planning defined by PM
	Evaluating and Reporting	<ul style="list-style-type: none"> Report test progress to QC lead and PM 	<ul style="list-style-type: none"> Evaluating test results and progress Write test summary reports based on the information gathered during the test
UAT	UAT	<ul style="list-style-type: none"> Verify Customer Feedbacks or Change Requests (CRs) Perform re-testing and regression testing after changing the source code Update test specifications 	<ul style="list-style-type: none"> Follow-up Controlling
Maintenance	Maintenance	<ul style="list-style-type: none"> Review and contribute to tasks Create/update test specifications Execute tests for these tasks Perform regression testing 	<ul style="list-style-type: none"> Review and estimate time for tasks Follow-up test progress

Purpose	Tool
Website Performance Index	Google PageSpeed, Webpagetest, JMeter
Security Tool Scan	SonarQube, ZAP
Issue Management System	Jira
UI Testing	Perfect Pixel, BrowserStack
API Testing	Postman, SoapUI
Automation Testing	Selenium

QC Tools and Platform

Similar to the tools we identified for the development process, we have selected a handful of choice solutions and platforms that we often utilize for quality-control testing.

Website Performance Index

Google PageSpeed reports on the performance of a page on both mobile and desktop devices and provides suggestions on how that page may be improved. WebPageTest is a web-performance tool that uses real browsers to access web pages and collect timing metrics. Finally, Apache JMeter is an Apache project that can be used as a load-testing tool for analyzing and measuring the performance of a variety of services, with a focus on web applications.

Security Tool Scan

SonarQube and ZAP were introduced in the development process, but we are also including it here with more information about how it is involved in the QC process.

SonarQube is also used for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities. OWASP ZAP (Zed Attack Proxy) is intended to be used by both those new to application security, as well as professional penetration testers. Some of the built-in features include intercepting proxy server, traditional and AJAX Web crawlers, automated scanner, passive scanner, forced browsing, Fuzzier, WebSocket support, scripting languages, and Plug-n-Hack support.

UI Testing

PerfectPixel allows developers and markup designers to put a semi-transparent image overlay over the top of the developed HTML and perform a pixel-perfect comparison between them.

BrowserStack is a cloud web and mobile testing platform that allows developers the ability to test their websites and mobile applications across on-demand browsers, operating systems, and real mobile devices.

Automation Testing

Selenium is composed of several components (Selenium client API, Selenium WebDriver, etc.) with each taking on a specific role in aiding the development of web-application test automation.

API Testing

Postman is the collaboration platform for API development. Postman simplifies each step of building an API and streamlines collaboration so you can create better APIs.

SoapUI is an open-source web service testing application for Simple Object Access Protocol (SOAP) and representational state transfers (REST). Its functionality covers web-service inspection; invoking, development, simulation, and mocking; functional testing; load and compliance testing.

Planning + Governance

ADOBE COMMERCE IMPLEMENTATION PLAYBOOK

04

Getting to what's right

Like the critical need to understand the full scope of the project before beginning, having a plan for delivering a solution on time, on budget, and that meets your needs is instrumental to success. This requires both identifying a foolproof blueprint for the design and implementation of the solution and executing that blueprint while measuring for success.

It is a commitment from the entire team that begins at the initial ideation and doesn't end until well after launch. And while this is a detail-oriented and sometimes tedious aspect of the process, the work put into the plan and governance will make any obstacle faced during the project much easier to navigate around and overcome.



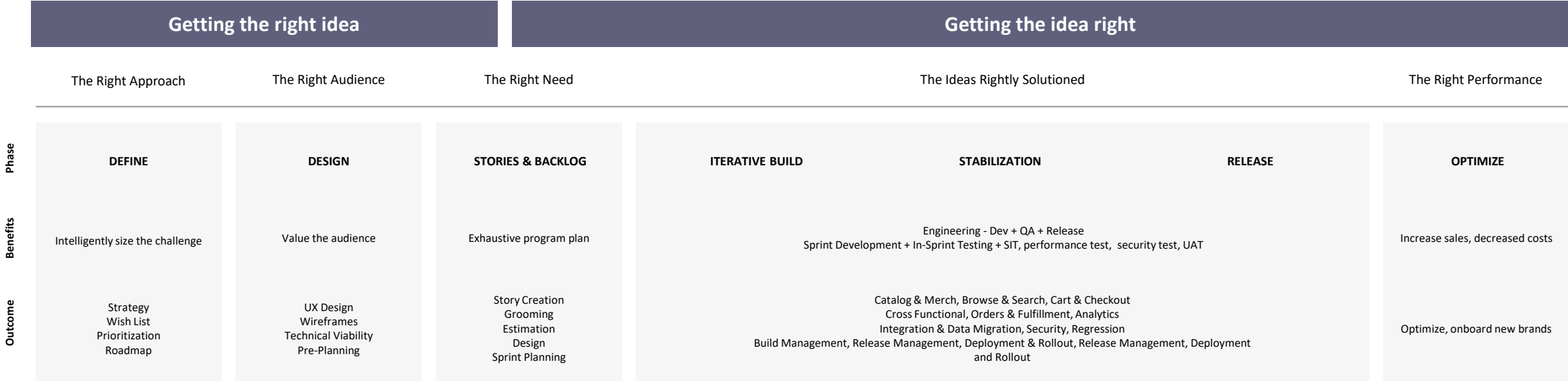


Delivery planning and approach

Strategic commerce projects can be delivered with several approaches and different project management methodologies. However, for a successful delivery and optimal execution, a hybrid-agile approach is recommended.

This approach consists of the following steps:

- “Define” phase to plan, architect, and strategize
- Agile sprints to build and design the solution
- Integration, performance, and user-acceptance testing for pilot
- Scrum team rollouts for brand and commerce sites



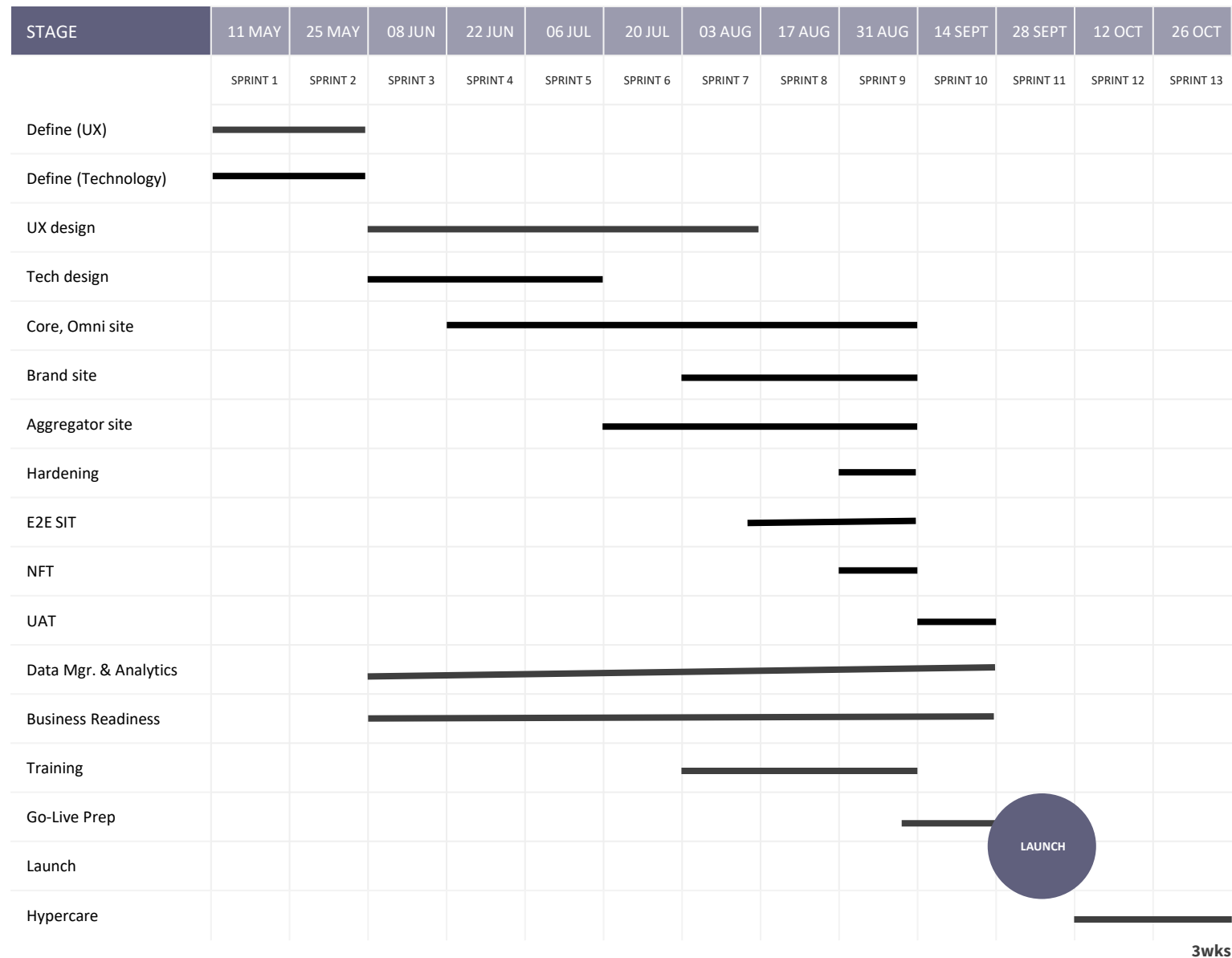
The hybrid model allows us to leverage the power of agile methodology, keeping on target for the timeline, cost, and quality. The entire plan is divided into two distinct phases, followed by the launch, optimization, and maintenance activities.

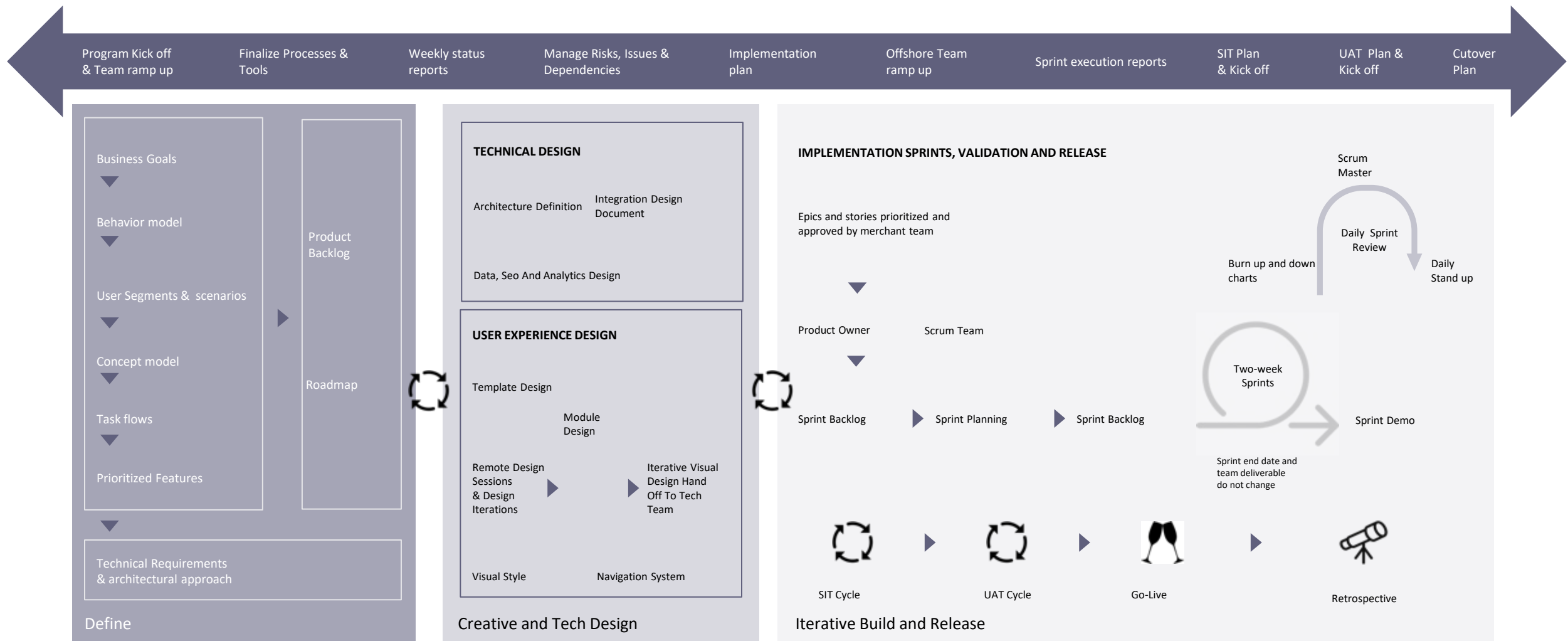
1. Getting the right idea:
This is a pre-implementation phase to ensure that the requirements, design, and technology solution are strategized, documented, and collaborated across stakeholders, the backlog for implementation is groomed and built, development sprints are planned, and a sequential strategy for building the product is finalized.

2. Getting the idea right:
This is the implementation phase to build the product iteratively, test thoroughly, and release appropriately. The first step is to develop the code through multiple sprints; each sprint contains in-sprint development and testing, followed by system-integration testing, performance and security testing, user-acceptance testing, and release activities.

Project planning is a critical activity. It guides the stakeholders, teams, sponsors, and project managers on how to monitor, manage, and control project phases. It also helps identify the goals, avoid missing deadlines, prioritize essential tasks, reduce risks, and deliver desired results.

The project plan usually consists of four phases: initiation and planning, execution, launch, and post-launch. Each phase can be delivered through multiple smaller sprints, thereby managing progress and outcomes through smaller increments and with a higher degree of agility. Planning is unique, however, in that it is usually a part of the other three phases. We have provided a detailed breakdown of each phase and an example of a project plan for a successful commerce implementation:





The approach we have defined shows different project phases coming together to enable successful project execution. The PMO processes and a close oversight at various levels ensure success at each phase (and sprints) building up toward a successful, timely execution with optimal quality and cost.

RACI

While executing a big project, everything often seems to be going fine at first. The project roadmap is detailed and comprehensive with clear milestones. Resource plans and stakeholder commitments are in place. However, soon things may start falling apart with missed deadlines, forgotten tasks, etc. The project schedule and plan fall off from that point.

This often happens due to a lack of clarity on the project's roles and responsibilities. Clear roles and responsibilities are a critical success factor for any project. Without them, projects can quickly derail and stakeholders are left trying to manage through exceptional methods with time/budget overrun.



A RACI (Responsible, Accountable, Consulted, Informed) matrix—like we mentioned in the project scoping section—is a useful tool that defines and documents ownership and responsibility at each stage of the project. Within the matrix, every task, milestone, and decision in the project is mapped out to specific roles or individuals. It helps project management to streamline the processes by ensuring each team member and stakeholder understands their specific roles. This should be defined in the planning phase of the project and accepted by each stakeholder.

How do we collectively collaborate?

RASI Framework
 Responsible
 Approver
 Consulted
 Informed

Activity	Responsible	Approver	Consulted	Informed
Program and Project Planning	●		●	●
Product Decisions and Prioritization	●		●	●
Managing Non-Platform Dependencies	●			
Requirement Definition / Gathering	●		●	
Scope determination /Finalization of Scope	●	●		
Solution finalization	●	●		
Identify the dependencies	●		●	
Technical / Solution Design	●	●		
Creating Branding guidelines	●			●
Visual Designs	●			●
UX Designs	●			●
Wireframes	●			●
Story Grooming	●	●	●	
Sprint Release Planning	●	●		
Code, Code Review and Unit Test	●			
Functional Testing	●			●
Test Data Creation	●			●
Test Payment Accounts / Credit Cards	●			●
System Integration Testing	●		●	
Change Request Process	●	●		
Security/Penetration Testing and Logging Defects in Jira	●			●
Security/Penetration Testing - Bug Fixing	●			
Performance Testing and Logging Defects in Jira	●			●
Performance Testing - Bug Fixing	●			

● Service Provider ● Customer




Project governance

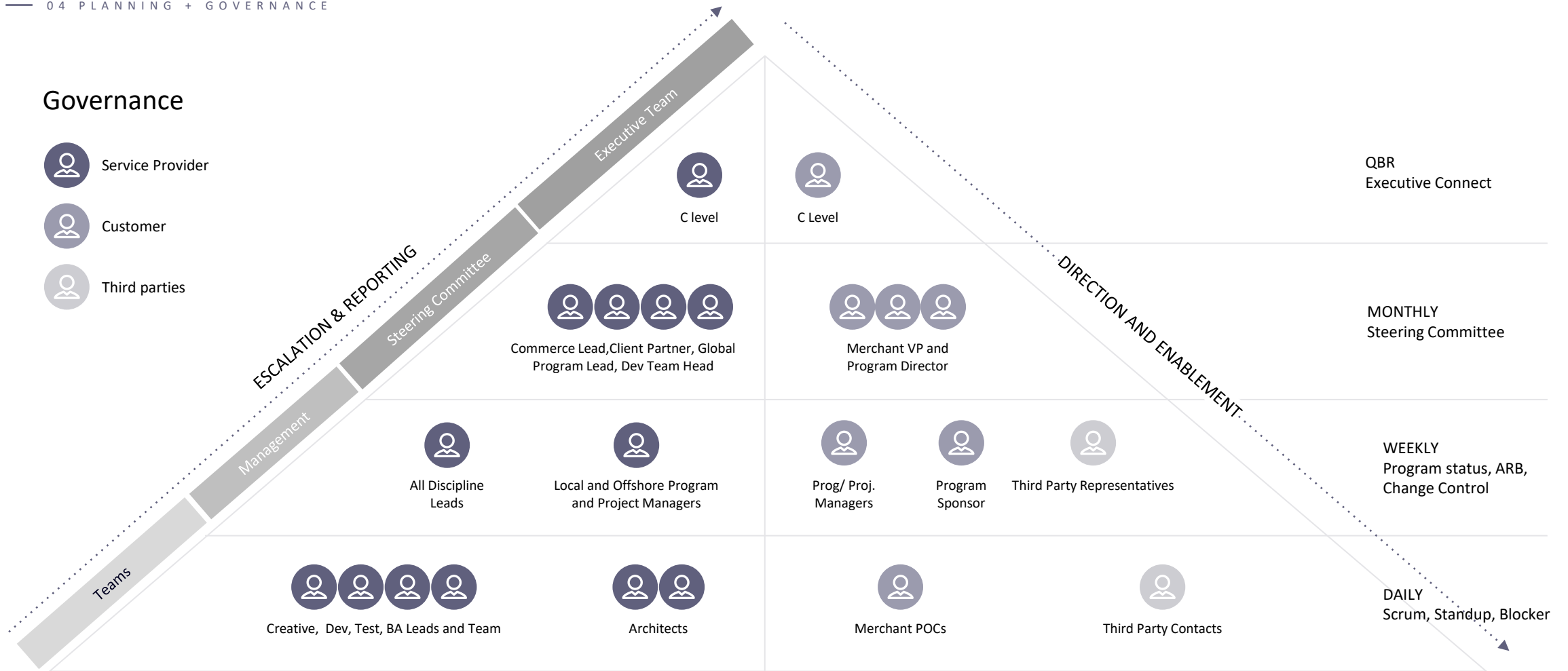
Project governance is an oversight function aligned with the organization's governance structure and that encompasses the project life cycle. It provides the project manager and team with structure, processes, decision-making models, and tools for managing and controlling the project, while also ensuring the successful delivery of the project. Project governance is a crucial element, especially for complex and strategic projects.

The governance model defines, documents, and communicates custom and effective practices to provide a comprehensive method of controlling the project, providing periodic visibility at every level to ensure success. It contains a framework for making decisions; defines roles, responsibilities, and liabilities for the accomplishment of the project; and governs the effectiveness. The governance structure rolls up from the execution team all the way to executive management, defining the activities, reporting, escalation, and information flow.



Governance

-  Service Provider
-  Customer
-  Third parties



At various levels, the teams look into specific sprint and project metrics to understand the progress and take corrective actions as necessary. These sprint-level metrics can include the velocity and burndown of each sprint.

Regular meeting details

Quarterly Business Review

- Discuss growth escalation strategies
- Highlight current success & goals
- Align on desired outcomes for the coming quarters

Monthly Steering Committee

- Coordinate and review project progress
- Decision making on major impact items (if any)
- Dentsu ensures customer satisfaction and concerns are recorded and addressed

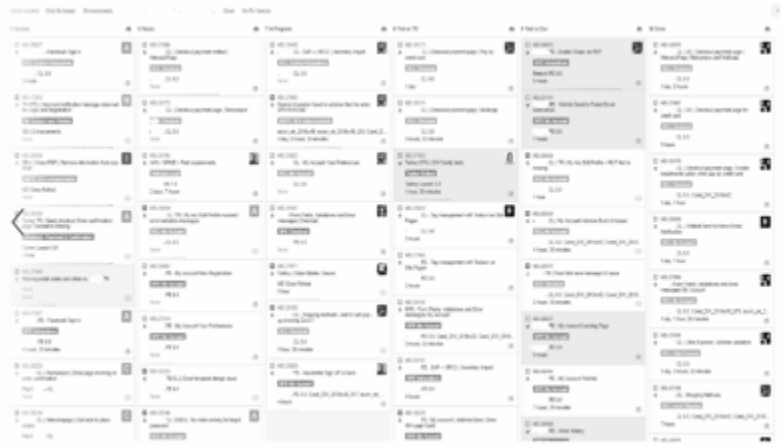
Weekly Project Committee

- Decide on objectives, plan, organization for the week
- Make architecture decisions as needed
- Review & act on project status reports
- Demos the platform and features
- Escalate requests/issues/suggestions

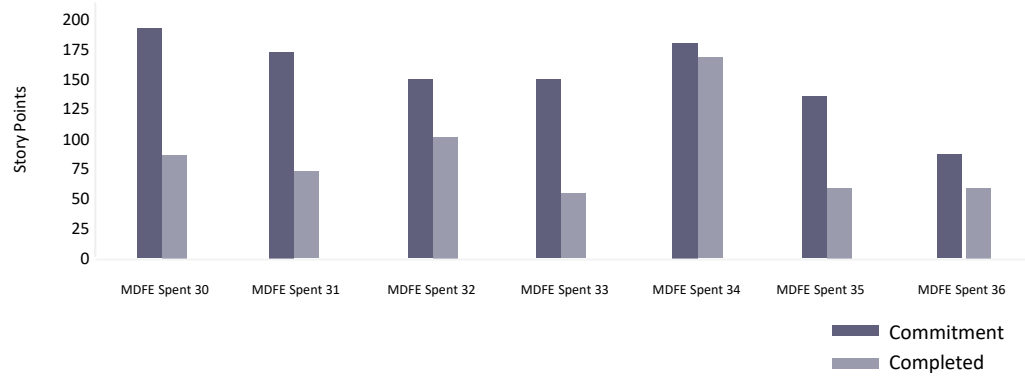
Daily Meeting

- Discuss and follow up on action items, including current sprint/boards/outstanding tickets
- Monitor project progress

Storyboard and Velocity

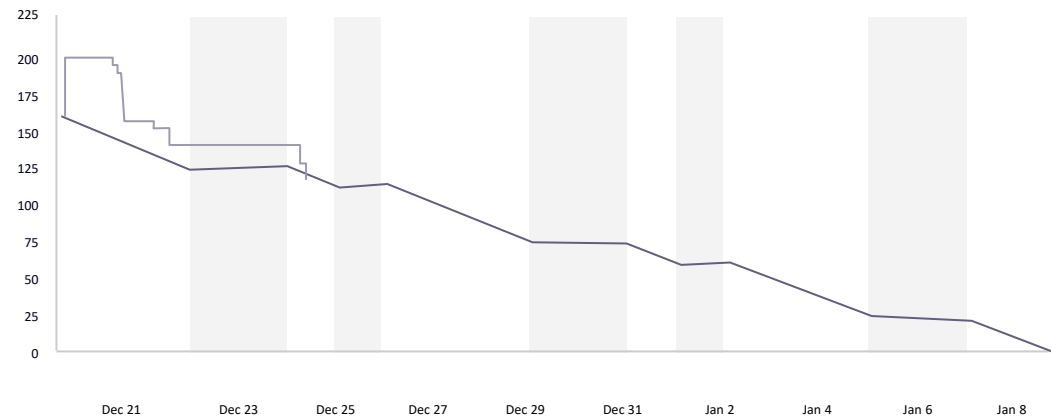


MD-FE Delivery
Velocity Chart

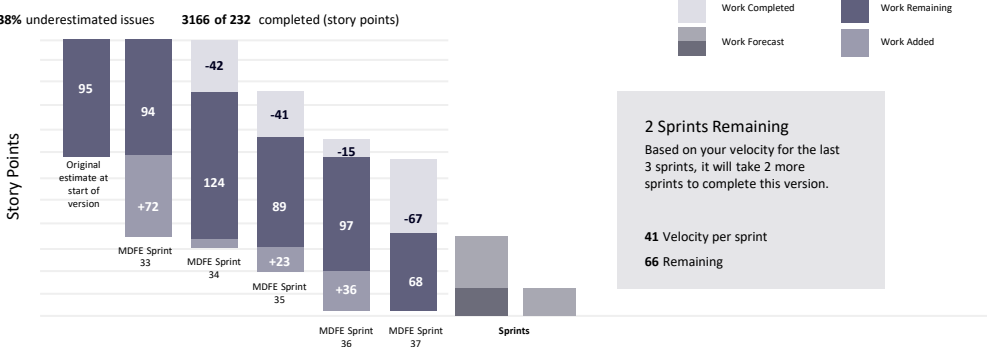


Sprint and Release Burndown

MD-FE Delivery
Release Burndown



MD-FE Delivery
Release Burndown



Apart from the sprint metrics, it is also essential to measure project and quality performance KPIs. Not only does this help to ensure the level of quality throughout the plan, but it keeps the team on track and avoids the project from going off the rails.

Challenges or changes will occur throughout the duration of the project. Empowering the right people within your organization with the ability to track, measure, and pivot when a challenge is met is going to increase the likelihood that you come out of the project having met your goals and are satisfied with the result.

Key Performance Measurement	Unit of Measure	Reported Metrics
Test Coverage	%	# of testable requirements covered by test cases VS Total of baselined testable requirements
Defect Density	%	# of valid Defects found VS Total of Test cases executed
Defect Leakage to SIT/ UAT / Production	%	Defects reported in Production VS Defects reported in Production + Defects reported by QA+UAT
Test Effectiveness	%	Valid Defects raised/ Valid Defects raised Rejected defects
Code Quality	# + %	Complexity, LoC, Violations, Code coverage for the sprint

Architecture + Integrations

ADOBE COMMERCE IMPLEMENTATION PLAYBOOK

05

The platform of choice

Adobe Commerce is one of the most trusted and reliable ecommerce platforms on the market. And that trust and reliability has been proven in the growth of the businesses that use it. According to [Hosting Tribunal](#), brands leveraging Adobe Commerce as their ecommerce platform grow three times faster compared to brands on other platforms.

Adobe Commerce's ability to handle any complexity your brand might face has made it a popular choice as companies enhance their sophistication. Whether brands are focused on a single site in one region or need to connect multiple sites or stores across the globe, Adobe Commerce has the infrastructure to create a seamless and connected experience. As a partner, we then leverage our expertise to customize the site to fit your unique needs.





Adobe Commerce capabilities

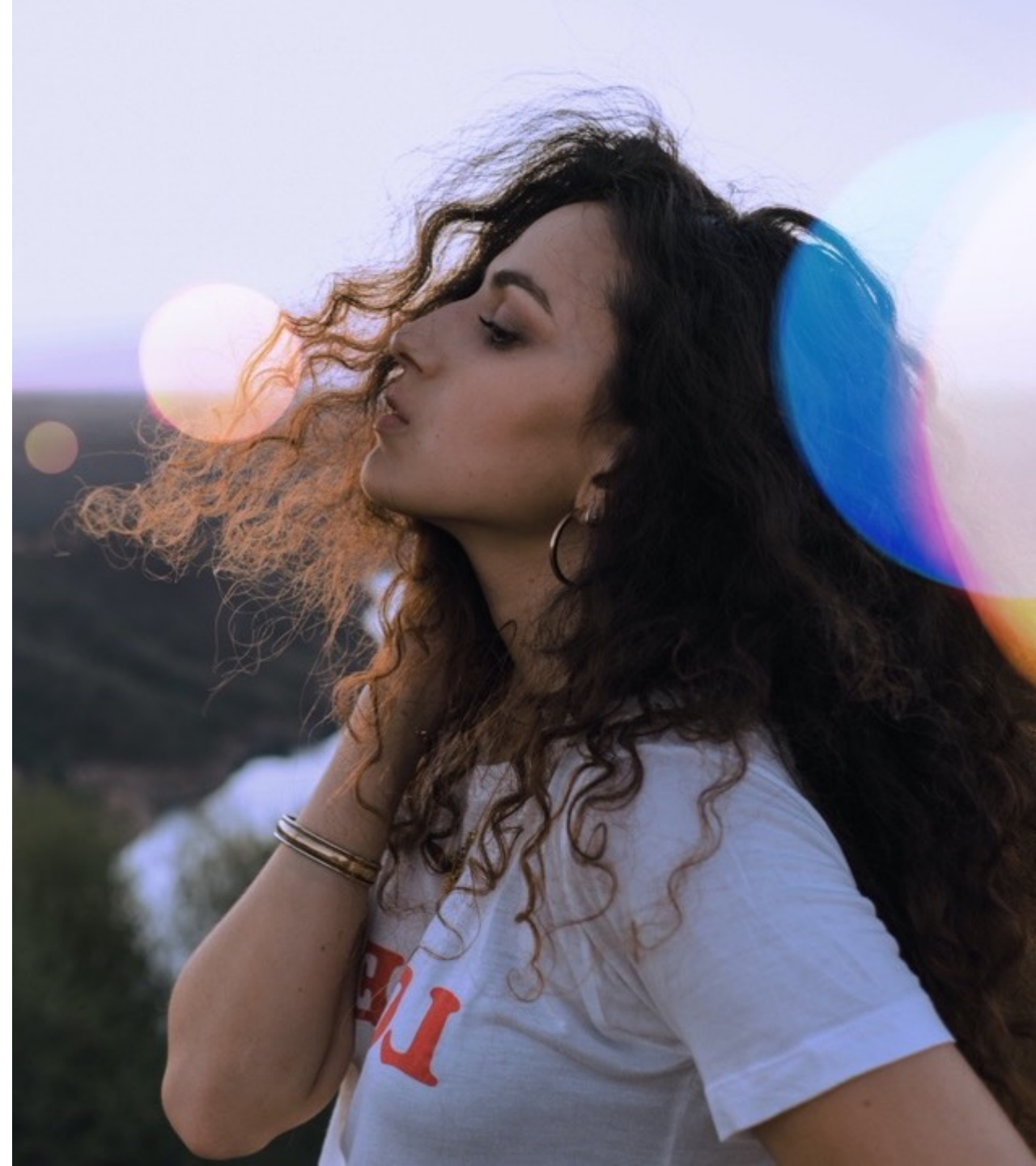
Adobe has been well known from creative products, but it also has a significant digital business providing content management, analytics, campaign orchestration, account-based marketing personalization capabilities, and, of course, commerce.

But what makes an experience good and differentiated from competitors?

- It needs to be rich and engaging
- It should be driven by data and contextual for the visitor
- It needs to be intelligent
- It needs to be seamless across touchpoints and wherever your visitors are

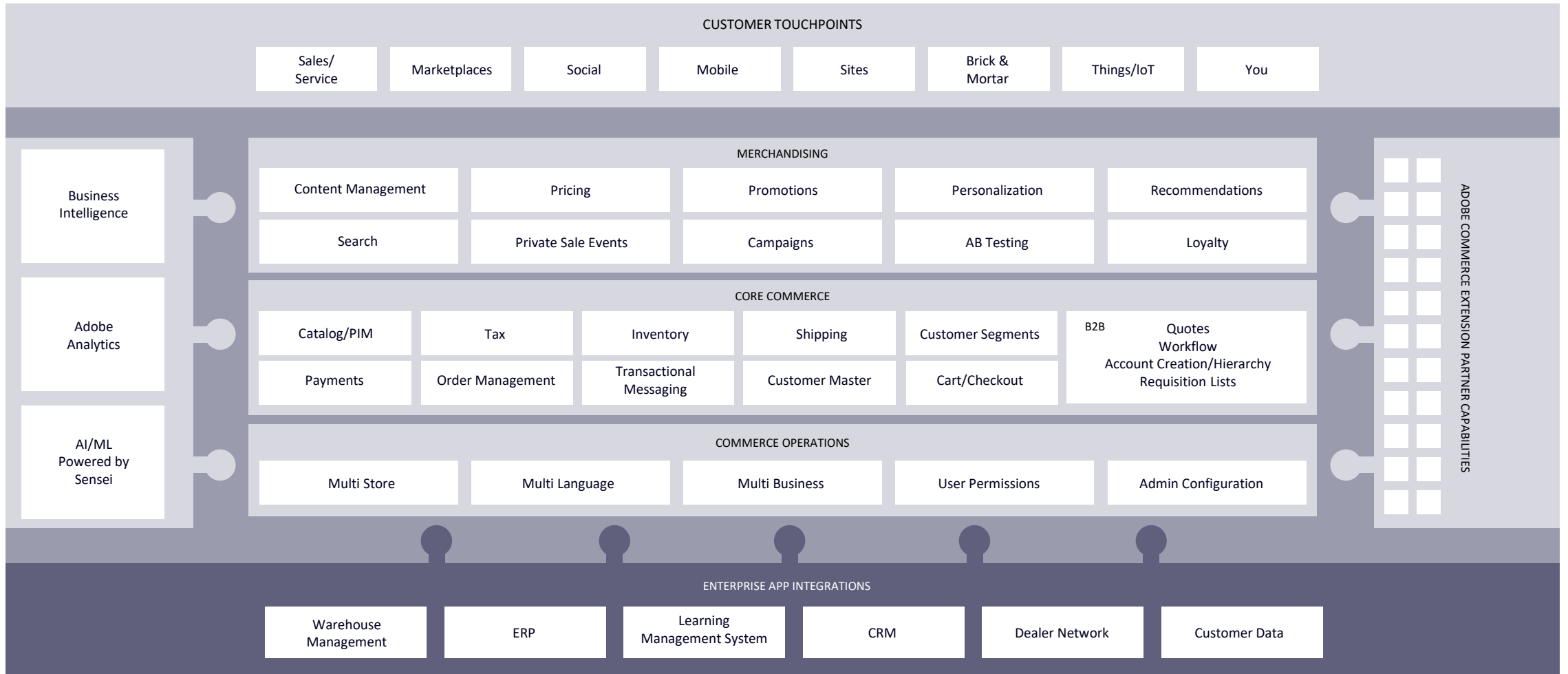
To build great shopping experiences, enterprises need powerful, modern platforms that are open, extensible, scalable, and secure, not to mention modern tooling for agile development. Adobe Commerce is the perfect platform to fill all of those needs.

Adobe Commerce is a fully featured, market-tested ecommerce platform that's built to deliver rich, personalized commerce experiences across every channel and business model. Adobe Commerce can serve as either an end-to-end commerce application or a reference application within a larger business landscape. It is modular and flexible with an extensive API built in REST and GraphQL and it can be used with a variety of heads—anything from WordPress and Drupal to AEM and others.



Adobe Commerce Capabilities Map

The Adobe Commerce application is made up of a collection of features that are self-contained modules



THIRD-PARTY MODULES	Commerce Features	Integrations	Payment Gateways	Shipping Providers
CORE B2B MODULES	Commerce Features	Quick Order	Requestion Lists	Quotes
CORE MODULES	Merchandising	Returns	Reward Points	Scheduled Import/Export
	Personalization	Staging & Preview	Page Builder	Related Products
	Search	Wishlist	Product Comparison	Reviews
	Inventory	Product Management	Customer Account	Cart/Catalog Price Rules
	Multi-site	Cart	Payments	Shipping

Adobe Commerce integration strategy

The ability to integrate your platform is “non-negotiable.” Companies want their ecommerce platforms accessible from a variety of touchpoints and seamlessly integrated into their technology systems, especially their ERP. Customizability, global scalability, and affordability also play a role in final platform purchase.

A holistic integration approach for both storefront and back-end systems are supported by performant GraphQL API, comprehensive REST API, and batch-file import between Adobe Commerce and other systems or services.

Adobe Commerce GraphQL API’s comprehensive storefront coverage is often chosen to integrate with other storefronts, including:

- Digital experience platforms (DXPs) like AEM, Bloomreach, etc.
- Content management systems (CMS) like Drupal, WordPress, etc.
- Modern custom storefront application like Adobe Commerce PWA Studio, Vue Storefront, etc.

GraphQL provides an efficient, channel-specific response, no over-fetching of data, and an agile deployment of new touchpoint features. It is also often chosen to integrate with sales channels such as mobile native apps, POS, IoT, social channels, and livestream commerce channels like Facebook, Google, Instagram, WeChat, TikTok, etc.



Adobe Commerce REST API's comprehensive coverage system configuration and data management features include product and catalog; cart, quote, and checkout; customers, account, and companies; and orders and returns. REST APIs support bulk operations, multiple authentication modes, and granular authorization, so REST APIs are often chosen to integrate with enterprise systems, including:

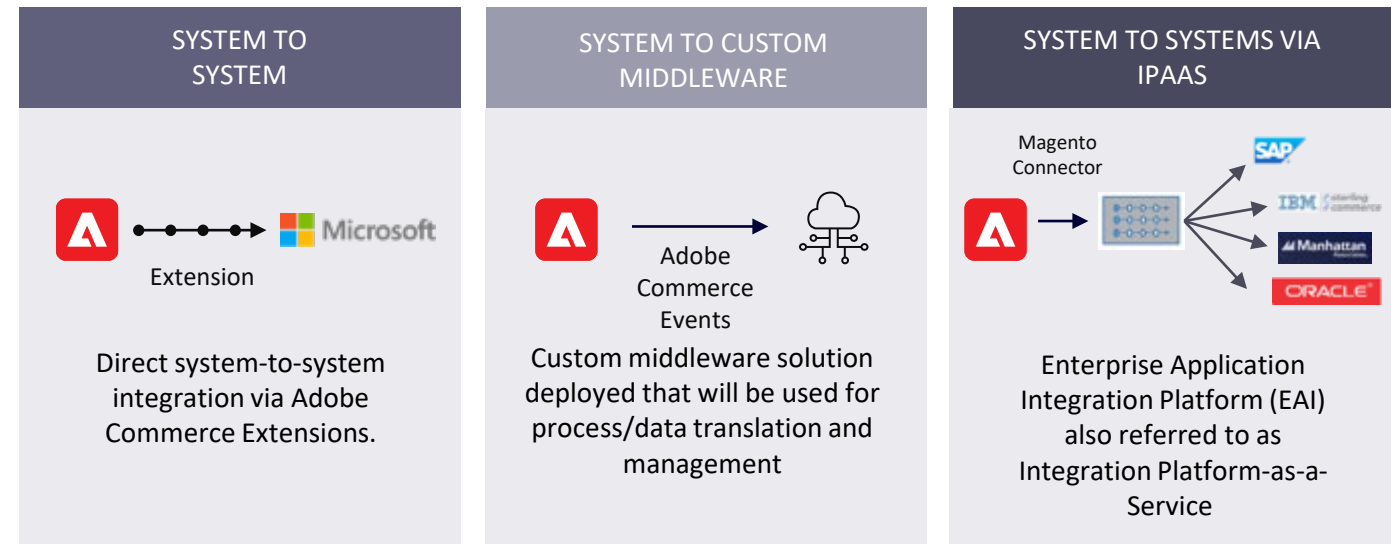
- Enterprise resource planning (ERP) systems like SAP
- Product information management (PIM) systems like Akeneo
- Customer relationship management (CRM) systems like Salesforce
- Order management systems (OMS) like MOM, Manhattan, and SAP
- Warehouse Management System (WMS) or third-party logistics (3PL) like Oracle, NetSuite, and SAP WM
- Configure, Price, Quote (CPQ) like Salesforce CPQ
- Digital Asset Management (DAM) like Adobe DAM.

Batch-file imports are also considered a good option to integrate enterprise systems like ERPs and PIMs, as that data does not change very often, and you often have better performance with scheduled file imports. So, batch-file imports are usually chosen for bulk data synchronization on daily/weekly bases and monthly full data synchronizations, whereas REST APIs are chosen for incremental data change synchronization, for better performance. This is also only considered a scheduled job but can be done frequently—potentially every 15 mins to 1 hour—depending on the business needs.



Adobe Commerce provide three flexible integration options:

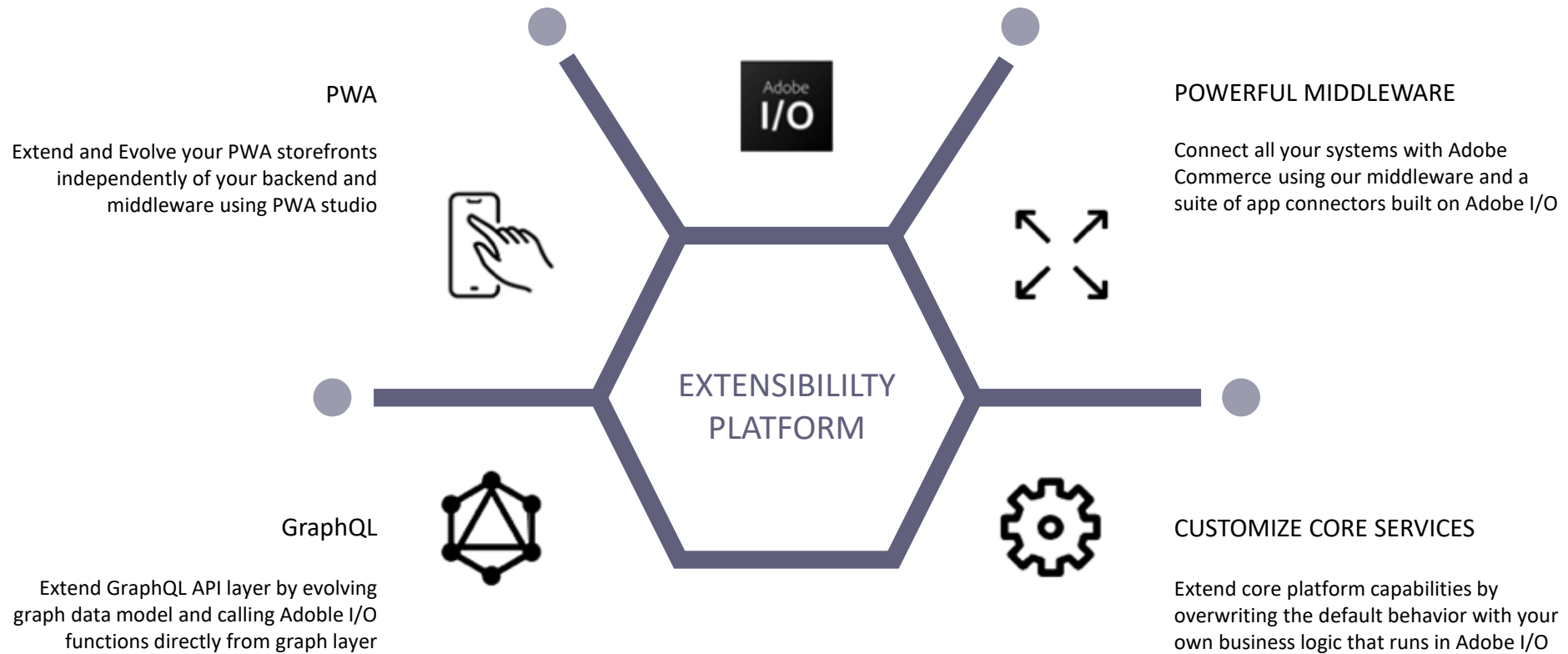
- Direct system-to-system integration with pre-built connectors. Some systems may already have Adobe Commerce extensions on Adobe Commerce Marketplace or their own website.
- System-to-system integration through custom middleware. The custom middleware solution deployed will be used for process data mapping, translation, and management.
- System-to-system integration via iPaaS (Integration Platform-as-a-Service), also referred to as EAI (Enterprise Application Integration Platform), e.g., Mulesoft, Boomi, Software AG.

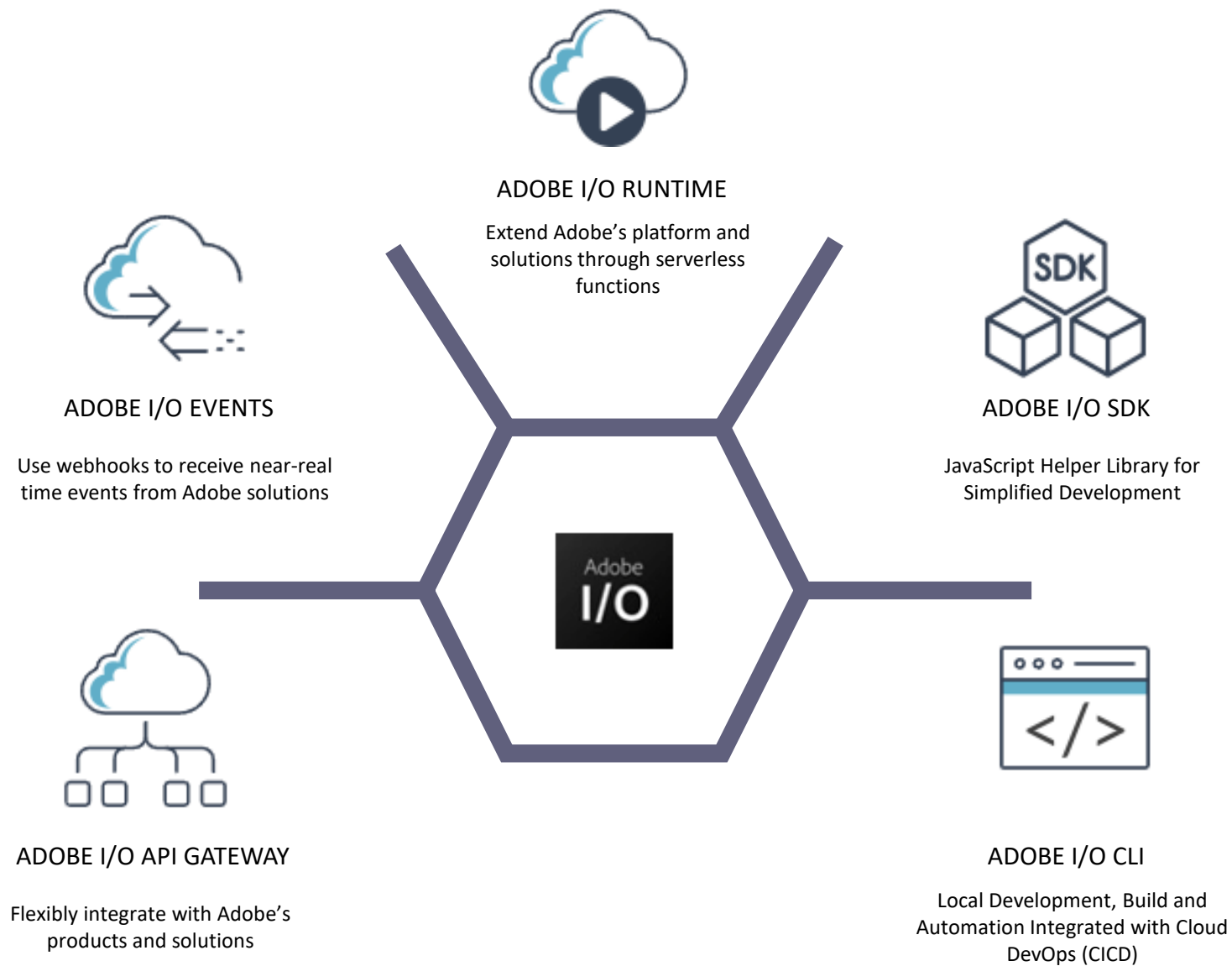


Even though real-time integrations are usually desired, it's not necessary for some scenarios. Adobe Commerce natively supports RabbitMQ as the message bus to enable asynchronous processes, which is recommended for some data that is not necessary to exchange in real time, but rather to update with batch-file exchange or REST batch data process API to process asynchronously.

Adobe Commerce Extensibility Strategy

Adobe Commerce's extensibility platform allows brands to efficiently customize processes, integrate systems, and deploy new capabilities while maintaining SaaS-like upgradeability.

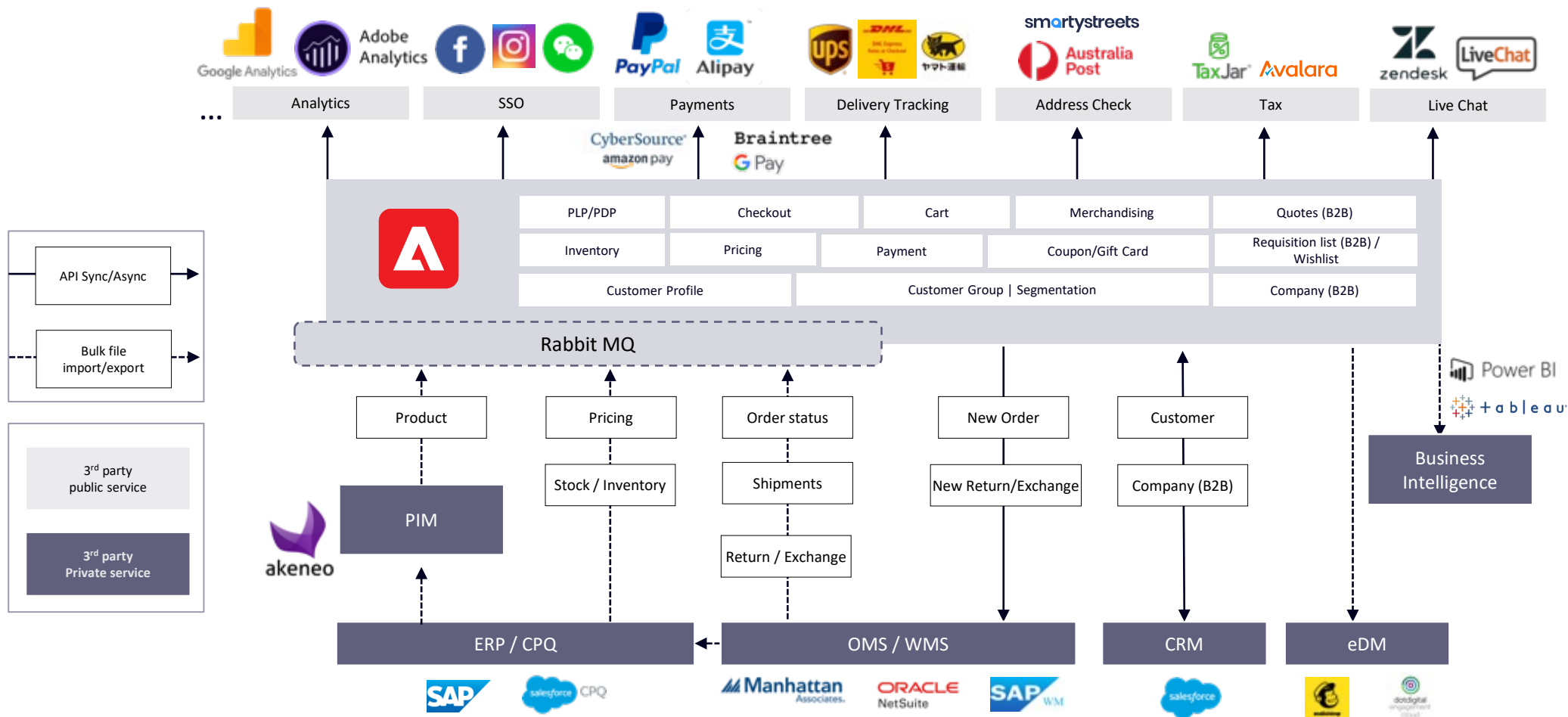




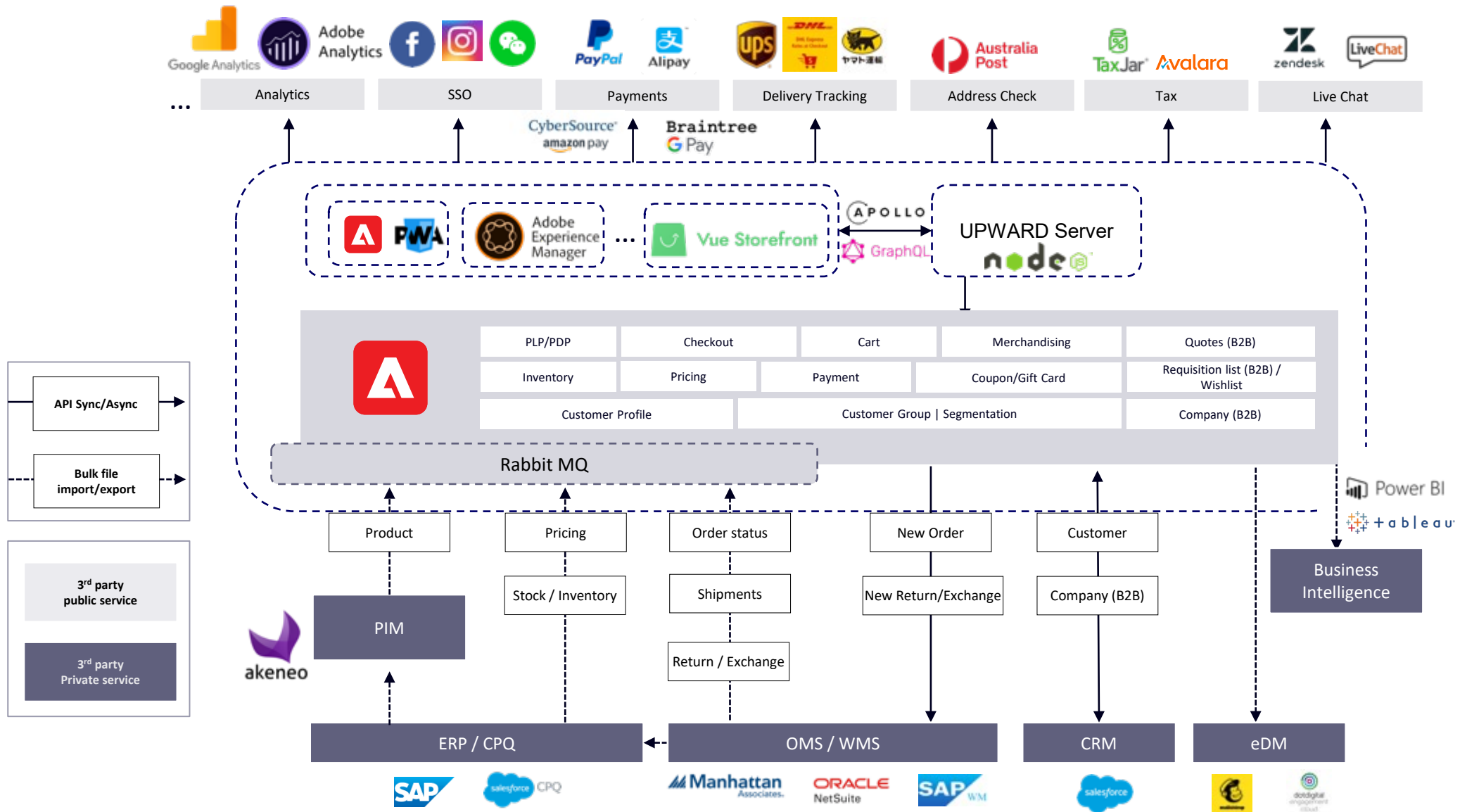
Typical integration point and dataflows

There are two main approaches to integrations and dataflows, which are very similar but have one key difference.

01 Monolithic approach: Adobe Commerce as storefront



02 Headless Approach: Adobe Commerce as the back end and integrated with DXP/CMS/custom application as storefront



The only difference between the monolithic and headless approach is storefront integration, which impacts the user experience for customer. Monolithic uses Adobe Commerce storefront directly to integrate with third-party services, while headless depends on its own storefront to customize and integrate to the same services. Some services like payment and SSO will need both storefront and Adobe Commerce customization to finalize the integration flow.

Some popular services already have great extensions to support Adobe Commerce or popular storefront solutions such as PWA Studio, AEM, and Vue Storefront, which can be found from their extension marketplace or from related third-party websites. Even if there is no existing extension, the effort to implement the integration between Adobe Commerce and other headless storefronts are similar—all third-party services usually have documents to explain how to integrate with them. Those service are just some examples; various countries and markets may have different choices.

For enterprise systems integrations, which are also usually called back-end integrations, there is an impact on the business data flow. Based on different business type and needs, it can use three different integration options, which we already introduced.





Product-mandatory data like SKUs and inventory and base prices usually come from ERPs, while sales prices are usually managed by each sales channel (e.g., Adobe Commerce) or CPQ (B2B or private sales). Because product-mandatory data (except inventory) does not change very often, best practice is to use scheduled batch updates, with REST batch update APIs or bulk-file import. For inventory, best practice is to have a full update on a daily basis for product inventory that is shared with different sales channel to avoid over-sales. Additionally, have incremental changes from your ERP scheduled within 24 hours.

Product catalog, metadata, and marketing content can be managed separately by each sales channel (e.g., Adobe Commerce), or from a central PIM. As metadata is also not changed frequently, best practice is to use scheduled batch updates, with REST batch update API or bulk-file import.

Order data includes order, quote(B2B), shipment, return, and exchange data that is usually managed from a centralized OMS and WMS system. Order data should be synchronized as soon as possible, so REST API is usually the best option. For better performance, consider reducing the number of API calls. For the order status, shipments, return, and exchange data, consider scheduling REST batch update APIs in hours or minutes.

B2B data is usually managed from a centralized CRM. A real-time API is used to verify existing customers and create new customers. For B2B, it might require introducing more APIs to synchronize different company employee, group, and price list between Adobe Commerce and your CRM or CPQ.

























There are some other system integrations like EDM for email marketing and business intelligence for business data analysis—which are usually done either via REST API or file export/import, which usually supported by existing extensions.



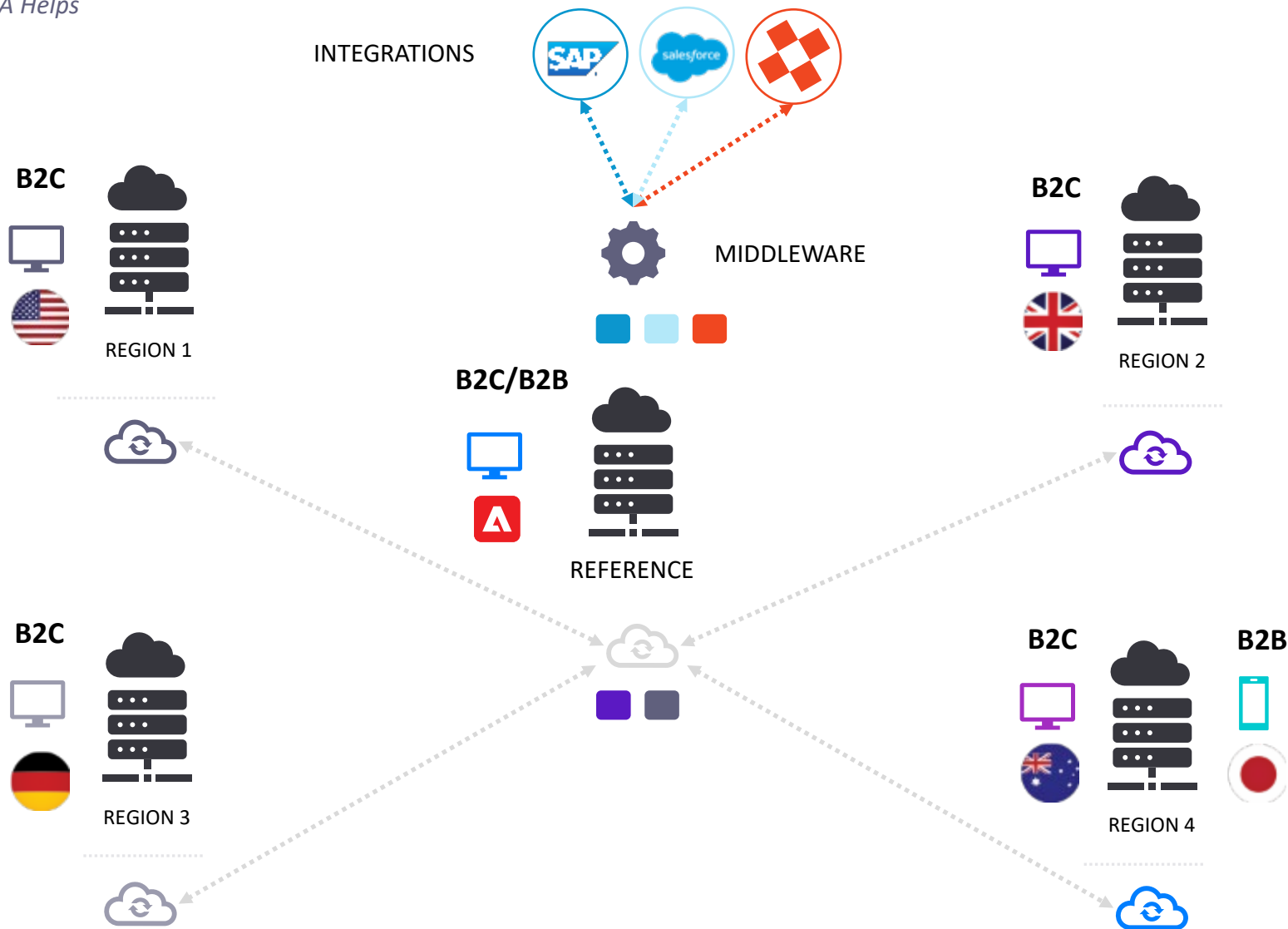
Global Reference Architecture (GRA)

When running businesses that have multiple sites for multiple brands in multiple local markets—with localized currencies, languages, media, shared catalogues and unique carts, and who want to avoid unnecessary costs for implementing same feature and integrations—GRA (Global Reference Architecture) is always a good option.

Without GRA

					COUNTING THE COST (POINTS OF DIVERGENCE)
MARKETS					<ul style="list-style-type: none">• May need unique themes
CHANNELS	B2C	B2C	B2C	B2C	<ul style="list-style-type: none">• May require channel-specific customization
INSTANCES					<ul style="list-style-type: none">• Separate environments• Separate upgrade paths/efforts
FEATURES					<ul style="list-style-type: none">• Each feature has to be maintained• Each feature may come from a different vendor
INTEGRATIONS					<ul style="list-style-type: none">• Each integration is a separate effort• Each integration may need unique changes
CODE REPOS					<ul style="list-style-type: none">• Different deployment schedules
DEV TEAMS					<ul style="list-style-type: none">• Different deployment processes• Different project teams

How GRA Helps



COUNTING THE COST (POINTS OF CONSOLIDATION)

- All themes share from a master theme—only unique elements need to be developed per market.
- Features and customization can be shared across markets and released with much less effort
- One master environment
- A much simpler upgrade path
- Features are consolidated per brand or market—only market-specific extensions are needed
- Only unique integrations need development—and only once.
- One master deployment process
- Multiple development teams governed by a master team

GRA is:

- An implementation approach
- A deployment strategy
- A process governance model

GRA is NOT:

- A product “feature”
- Unique to any commerce platform
- Only for global business use cases

GRA impacts:

How code is delivered

- Built around purpose-specific code repositories, which deliver different experiences.

How business systems are integrated

- Consolidates connections to business systems by brand and/or region.

How customization is developed and maintained

- Ensures customizations are centralized and domain-specific so that all customization effort is done from a holistic point of view for the business.

How new markets are enabled

- Simplifies the release of multiple channels and markets that would otherwise cost considerably more time and money.



Headless Commerce

Why Headless?

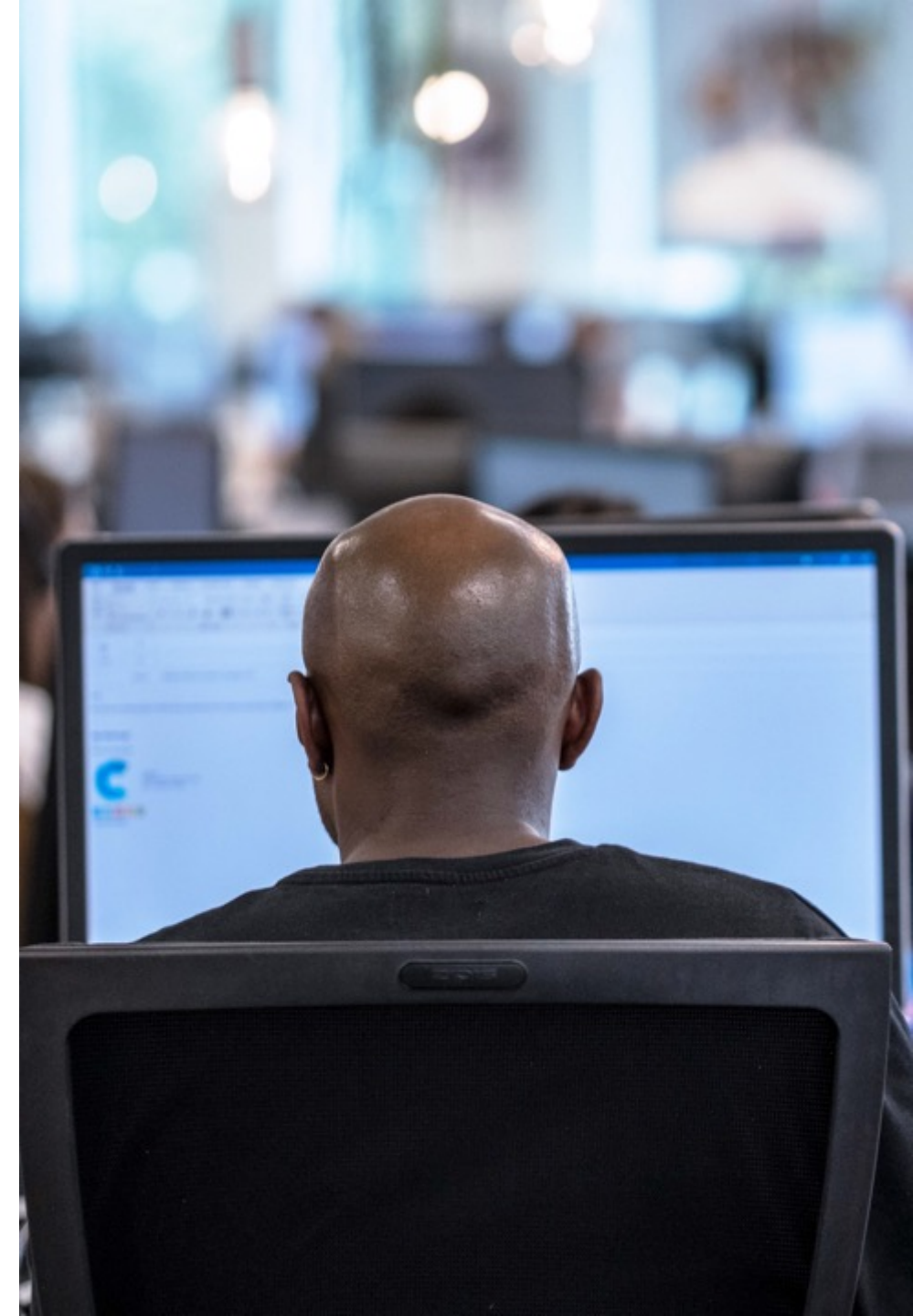
For starters, legacy enterprise commerce is expensive and difficult to scale due to siloes; legacy structures are reinforced through platform limitations; and innovation becomes challenging.

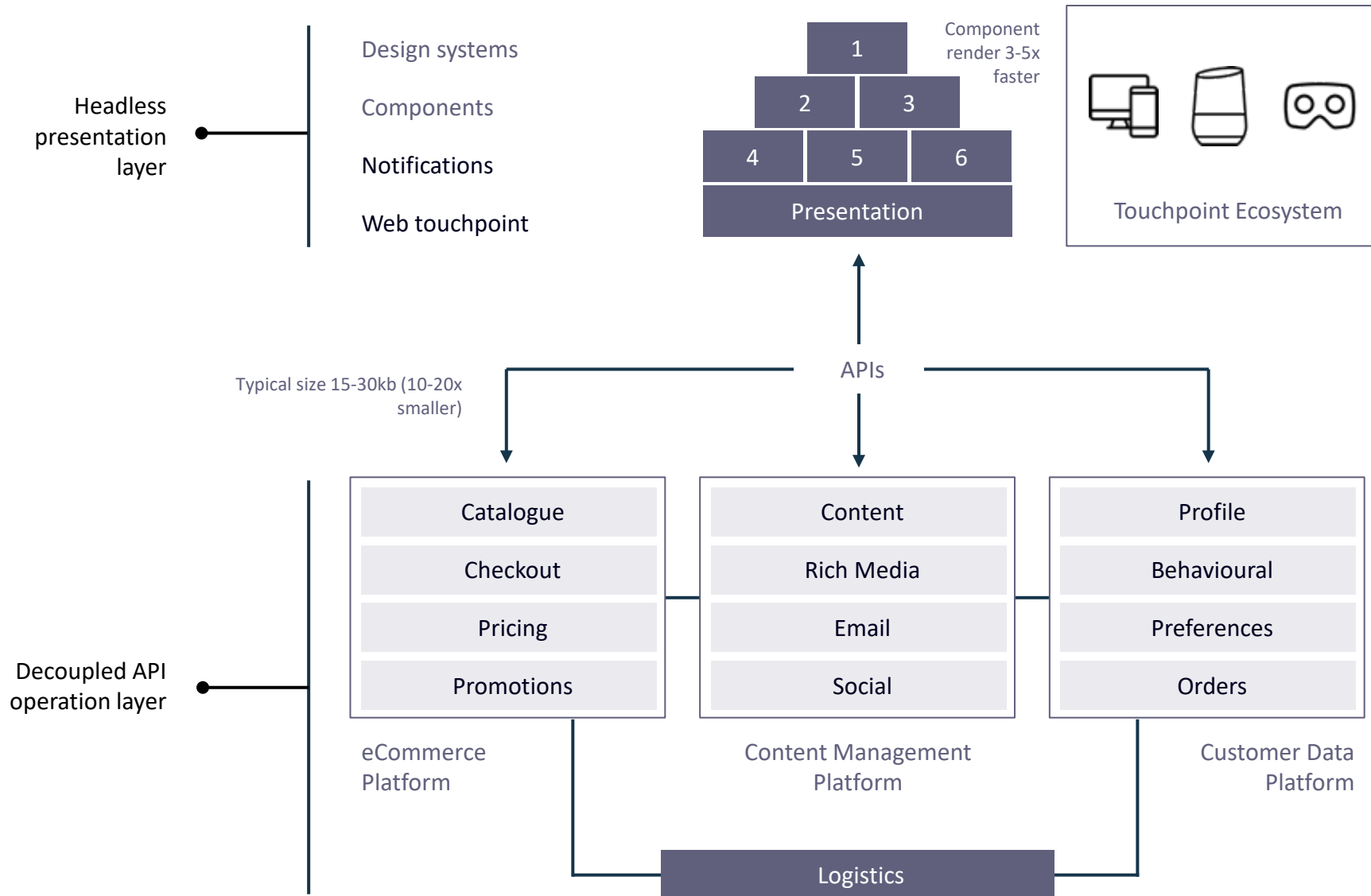
Customers are expecting a business to interact with them and engage with them across all channels, and customer-centric organizations are looking to build future-proofed platforms that can adapt to changing customer expectations.

Headless commerce is API-based commerce. It decouples business logic, as well as transactional and data aspects of commerce from presentation. Headless is an integrated framework that provides full flexibility for all channels and touchpoints, with a front-end experience layer that is separated from the platform itself. This allows brands the agility to deliver content such as products, data, and orders to any touchpoint both now and in the future, while being able to display it in any way they like.

Headless architecture is the technical separation of the head from the rest of the commerce application. Adobe Commerce is fully headless with a decoupled architecture that provides all commerce services and data via a GraphQL API layer. This architecture allows front-end teams to develop their front ends independently from Adobe Commerce, providing the agility to quickly build and test new touchpoints with emerging technologies.

Adobe Commerce's GraphQL APIs can also be extended with microservices that are deployed on Adobe's I/O Runtime. This provides unparalleled agility to integrate, extend, and customize omnichannel business processes without requiring code customizations to the back end, which ensures the core platform can be easily upgraded without impact to the front-end touchpoints. Adobe Commerce's GraphQL APIs are open sourced and part of our community engineering program with significant contributions and oversight coming from our community of developers.





Customer benefits of a headless architecture



Parity across touchpoints

All touchpoints, regardless of technology use the same API's and services resulting in consistent customer experiences across all channels and digital touchpoints



Futureproofed architecture

Even if merchants have no plans to build a custom head today, they have the piece of mind that their ecommerce platform is headless ready for tomorrow



Unconstrained control

Merchants have complete unconstrained flexibility to build any custom front-end experience, using any technology, at any time.



Innovate on new touchpoints

As new customer touchpoints merge, merchants have complete freedom to build and test new experiences without being constrained on waiting for their ecomm platform vendors roadmap or having to upgrade their backend



Reduce vendor lock in

Merchants can use a different vendor / technology for their front-end experiences and not be tied to the prescribed experience layer of their eComm platform



Separation of development activities

Changes to the front-end code can be deployed independently of any dependencies on the back end, allowing for CI/CD development principles and agility to allow for constant experience innovation



Improved scalability

Headless APIs are optimized for efficiency (only delivering exactly what's needed), high performance, and low latency.



Integrate commerce and content

A headless architecture allows merchants to inject commerce services into existing CMS-driven marketing sites and apps thus ensuring integrated customer experiences and re-use of existing technology investments

Adobe Commerce's Journey to Headless

2008 – 2016 (Magento 1)

- Monolithic architecture
- Sold as on-premise application
- Hosted/Managed by third-party hosting firms

2016 – 2018 (Magento 2.0)

- Monolithic architecture Launched “Magento Cloud” Managed cloud service hosted/managed by Magento on AWS Magento collaborates with Google on development of PWAs



2018 – 2019 (Magento 2.2 / 2.3)

- Separation of front-end / back-end architecture
- Development of GraphQL headless API's
- Launch of PWA Studio

2021+

- Decoupling of core services into isolated microservices architecture
- Evolution to hybrid SaaS architecture (e.g. Amazon Sales Channel, Product Recs)

2020 (Adobe Commerce 2.4+)

- Fully decoupled front-end/back-end architecture
- Completion of GraphQL API coverage
- Full reference PWA store (Venia) with completed components UI (Peregrine)
- PWA/Page Builder integration (business user control)

Headless and Microservices

One of the most important things to touch on quickly is not to get confused between headless and microservices. A lot of the time, we'll hear conversations about microservices in the same sentence as headless. They are completely different things. They can be used together, but they're completely different concepts.



MICROSERVICES <> HEADLESS

A microservices architecture is a term used to describe the practice of breaking up an application into a collection of smaller, loosely coupled services. Microservices enable individual back-end services to be:

- **Isolated from one another**
i.e. pricing service has no dependency on catalog service
- **Deployed a la carte**
Customers to deploy only the parts of the application they need
- **Scaled independently**
Resources can be assigned to high demand services, e.g. inventory lookup
- **Autonomously developed**
Can be developed and deployed independently of one another

Microservices really have nothing to do with chopping the head off the commerce stack or the APIs. When we think about those commerce services in the core code that are in the back office of Adobe Commerce, it's about isolating those services from one another. So, a microservices architecture is loosely breaking up all of those services like the pricing services, catalogue service, and inventory service, and making each one isolated from another. This ensures that they are a la cart so that you don't need to buy and deploy all of Adobe Commerce in one go.

Microservices can be scaled independently, and they are autonomously developed. So, microservices are very much akin to a multi-tenant SaaS development process. A lot of modern multi-tenant SaaS products are developed using a multi-service approach. And even Adobe's own SaaS products, when you think about Adobe Commerce Order Management being migrating toward a microservices approach, some of the new features that have been built into Adobe Commerce, like the new AI-driven product recommendations tool and other SaaS parts of Adobe Commerce, will be developed using a microservices approach. To be very clear, Adobe Commerce 2.4.x is not a microservices architecture but rather a headless architecture.



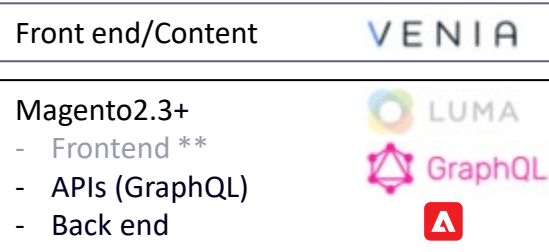
Evolution of Headless with Adobe Commerce

TRADITIONAL



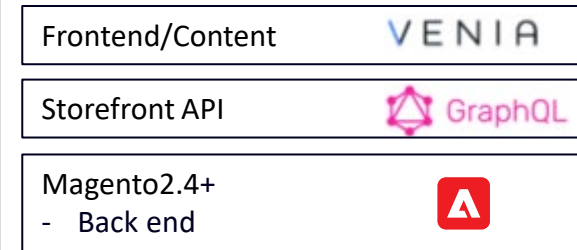
Monolithic approach (2015+)
PHP + UI Components
LUMA storefront/theme
Centralized content/authoring
Build, optimize and deploy

DECOUPLED

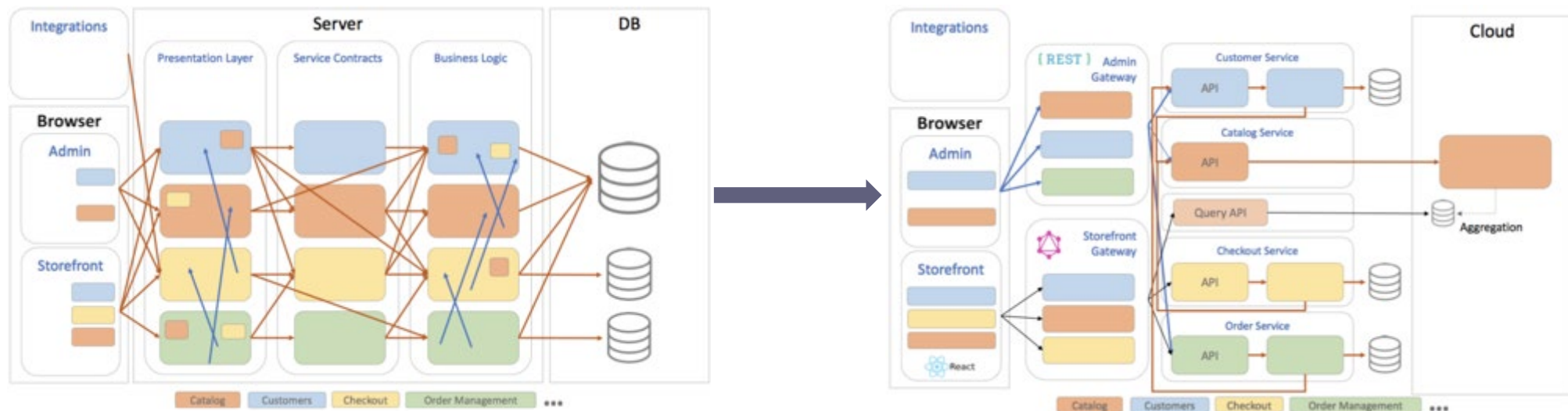


Semi-monolithic approach
PHP + UI Components or React
LUMA and Venia (PWA) storefronts
Decentralized content/authoring
Flexible build, optimize and deploy

HEADLESS



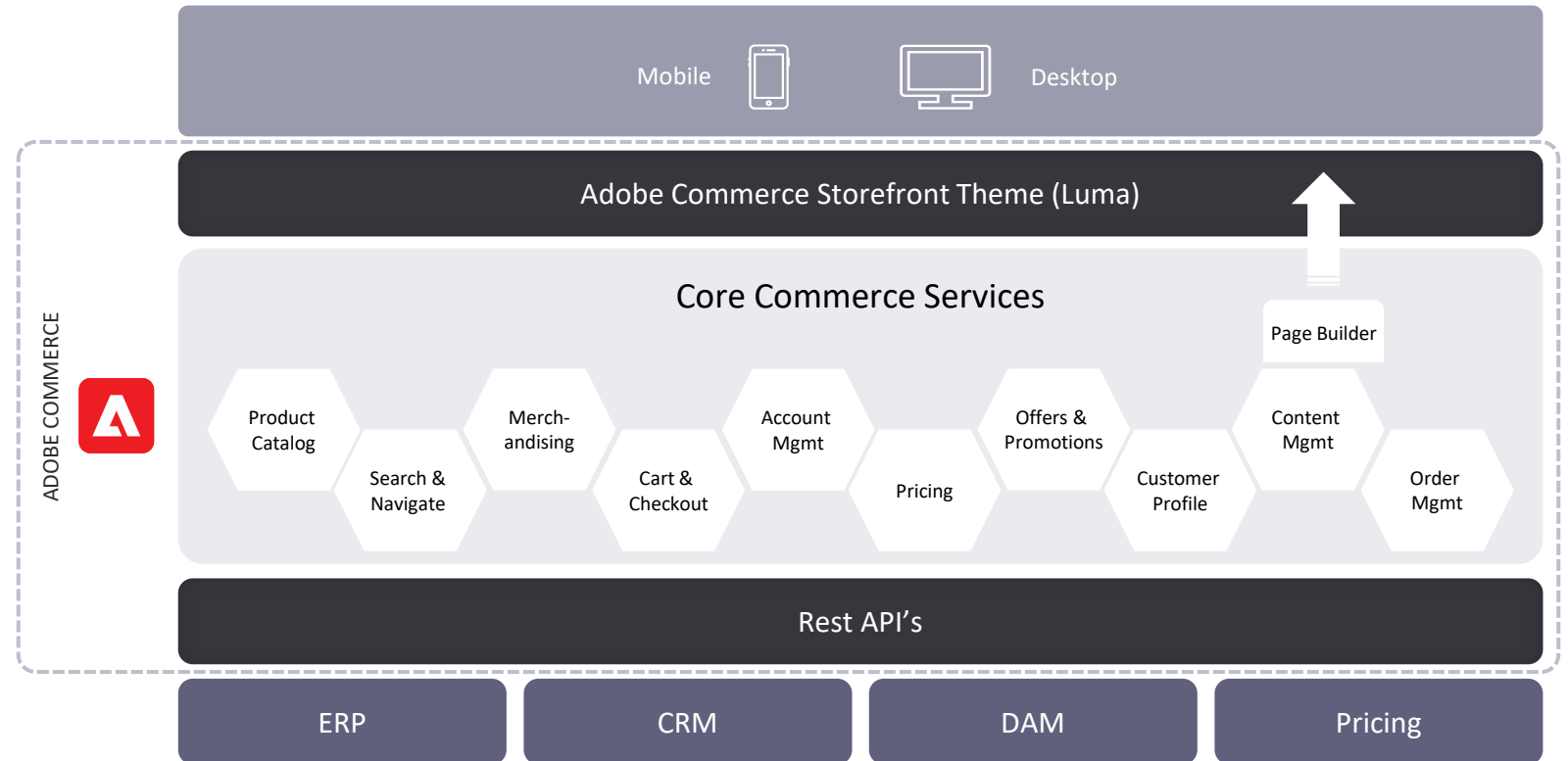
Modern headless approach
PWA + React
Venia storefront/theme (PWA)
More decentralized content/authoring
More flexible build, optimize and deploy



Coupled (legacy) Adobe Commerce Storefront Architecture

The current default deployment option for most commercial customers includes:

- 100% feature support across B2B & B2C
- Mature reference theme (Luma) that can be quickly deployed/customized
- Mature SI partner implementation expertise
- Fully compatible with commerce capabilities like Page Builder or Staging & Preview
- Broad compatibility with extensions in Adobe Commerce Marketplace



Cons of Legacy Storefront

- Not headless—all part of monolithic Adobe Commerce application—no separation of business logic and processes between front end/back end
- Not PWA—although theme is responsive, site performance lags far behind best-in-class PWA
- Front-end architecture (Adobe Commerce UI components)—Adobe Commerce/PHP specialists to build on legacy storefront

Before we get into headless options, let's start with the more familiar storefront architecture. If headless is decoupled, this would be the coupled storefront architecture, most commonly seen with our LUMA demos.

This is where the storefront capabilities are tightly integrated with the core commerce services, not separated by that GraphQL API layer. So, there is a lot of business logic coupled into that theme. This approach is not headless, and it's not PWA.

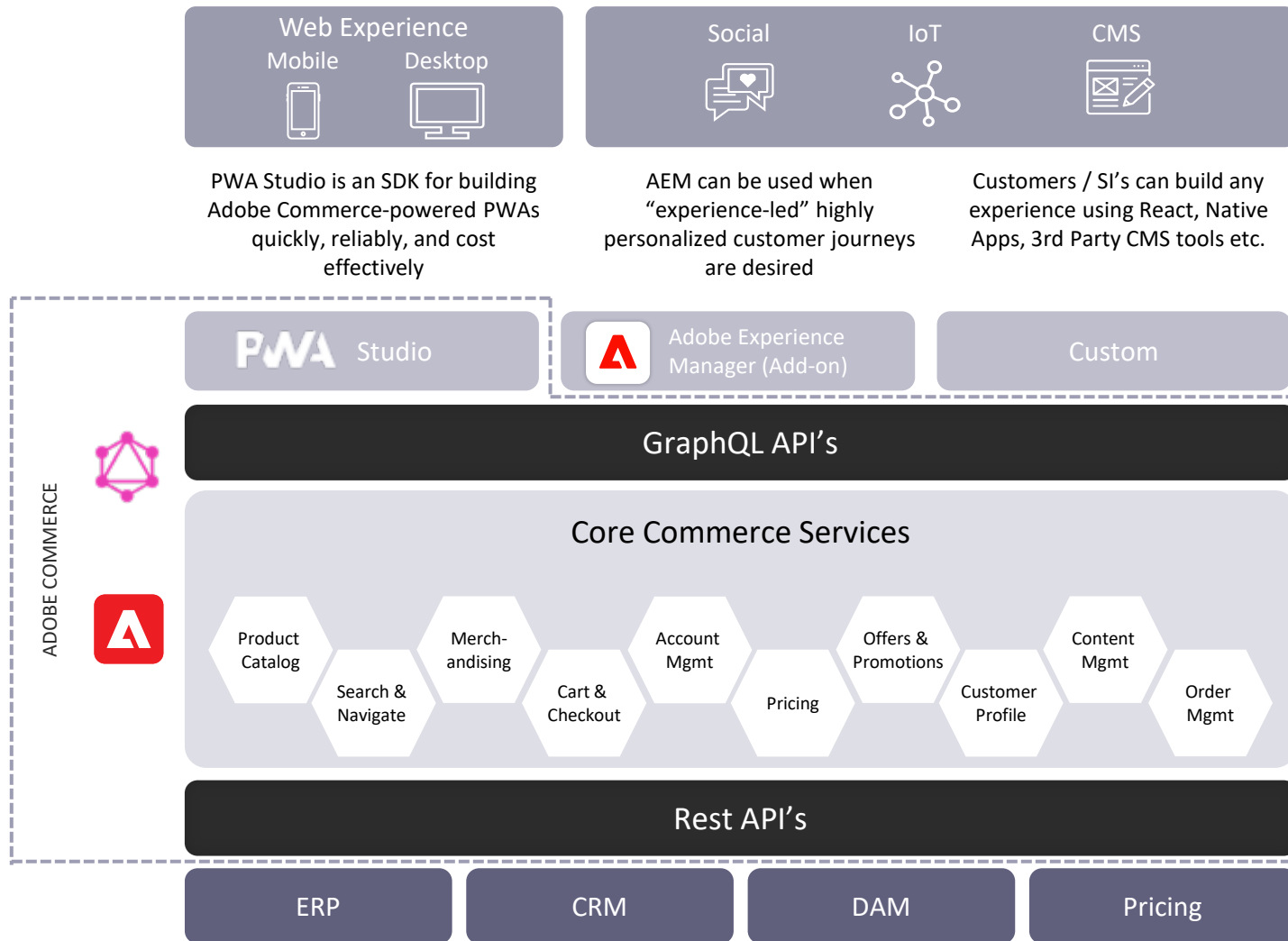
This is currently the most common option merchants use because it has 100% feature support with both B2B and B2C Commerce capabilities, the community is so familiar with it that there is a mature ecosystem of expertise around it, and it has broad compatibility with Marketplace extensions.

But it lacks the benefits we talked about earlier. Without separation of layers, there are many dependencies and potential points of failure when changes are made, it's not as performant as a well-implemented PWA, and if a merchant doesn't have expertise in Adobe Commerce or PHP development, they will have to go find those resources.



Headless Adobe Commerce Architecture

The benefit of Adobe Commerce's architecture is that it's not an all-or-nothing proposition and a merchant can find the right mix of solutions for their business. They can build a PWA Studio-powered PWA for their primary site experience or use AEM as they layer in complex customer journeys, all while building out a custom front end to experiment with new touchpoints. No other platform can match the time to market benefits and the flexibility that Adobe Commerce offers with its headless architecture.



Each approach is not mutually exclusive

Customers can build their own front end (head), use PWA Studio for web/mobile experiences and/or use AEM for the glass (in either a full or hybrid deployment).



Adobe Commerce has always allowed for headless deployments with REST APIs. And while REST is powerful, especially for bulk processing, when it comes to headless, GraphQL APIs enable faster development through an intuitive developer experience, greater flexibility by allowing for changes that don't impact existing APIs, and better performance by minimizing the amount of data retrieved to only exactly what is needed.

GraphQL is an industry standard for performant API, which is used by many of the top ecommerce platforms. That's a good thing since this means that it's a proven solution and expertise exists in the market.

While Adobe Commerce does have a coupled storefront as an option, it is by no means a required that a merchant use that Adobe Commerce legacy front end. A merchant can take advantage of Adobe Commerce's best-in-class commerce services to handle the back-end business processes and, using our storefront APIs, integrate their own decoupled storefront to drive the frontend experience.

Now, let's take a look at the various headless options.

Adobe Commerce Headless Storefront: PWA Studio

The first is a progressive web app built with PWA Studio. Part of this is enabled by the fact that a PWA is a headless storefront decoupled from the commerce back end. With PWA Studio, merchants can build high performing, reliable, and cost effective PWAs on top of Adobe Commerce to deliver best-in-class web experiences, both on mobile and desktop. As time goes on, this will overtake the coupled storefront as the default option.

Most merchants understand the direction that the industry is heading toward with regards to PWAs and many potential blockers are being removed rapidly.

Week over week, the number of partners building expertise in PWA Studio grows and we have an accelerating number of customer launches. The most recent update to PWA Studio included extensibility that will help make significant progress in compatibility with Marketplace extensions.

Many merchants may feel they aren't ready for headless and PWAs because they will have to rely so heavily on developers. One of the huge benefits of having both the commerce application and the head developed by Adobe Commerce is that it helps ensure compatibility across commerce capabilities.

In order to make PWAs more accessible and easier to manage for our merchants, we empower business users to manage day-to-day content changes, create new landing pages, and more using Page Builder. These two powerful capabilities together enable speed to market across all devices and experiences.

Adobe Commerce Headless Storefront: AEM (Adobe Experience Manager)

A powerhouse combo for your content and digital asset management needs, AEM helps merchants get personalized, content-led experiences into market faster, combining digital asset management with the power of machine-learning, Adobe Sensei-powered content and customer journey management.

Adobe Commerce plus AEM is a powerful story in that the commerce engine allows businesses to enable commerce through customer interfaces that are powered by AEM.

Adobe Commerce Headless Storefront: Custom Heads

The final option to discuss here is the option of building a custom front end. This option is for businesses that have existing expertise and in-house developer skills in a particular front-end stack like React and if they don't have developers skilled in Adobe Commerce's traditional front-end development, so they decided it was most cost effective to build their own custom React front end.

Naturally, this model requires strong customer or SI front-end development skills and resources, and you don't get the benefit of native compatibility with things like Page Builder that you get with PWA Studio. Any time a merchant is building something completely custom, they may lose time to market advantages.

But custom front ends also enable innovations and experimentation. There's a lot of talk about AR/VR or voice commerce, and an architecture like Adobe Commerce's allows merchants to explore these options without impacting their existing webstores.

Infrastructure + Deployment

ADOBE COMMERCE IMPLEMENTATION PLAYBOOK

06

Maximizing the Adobe Commerce platform

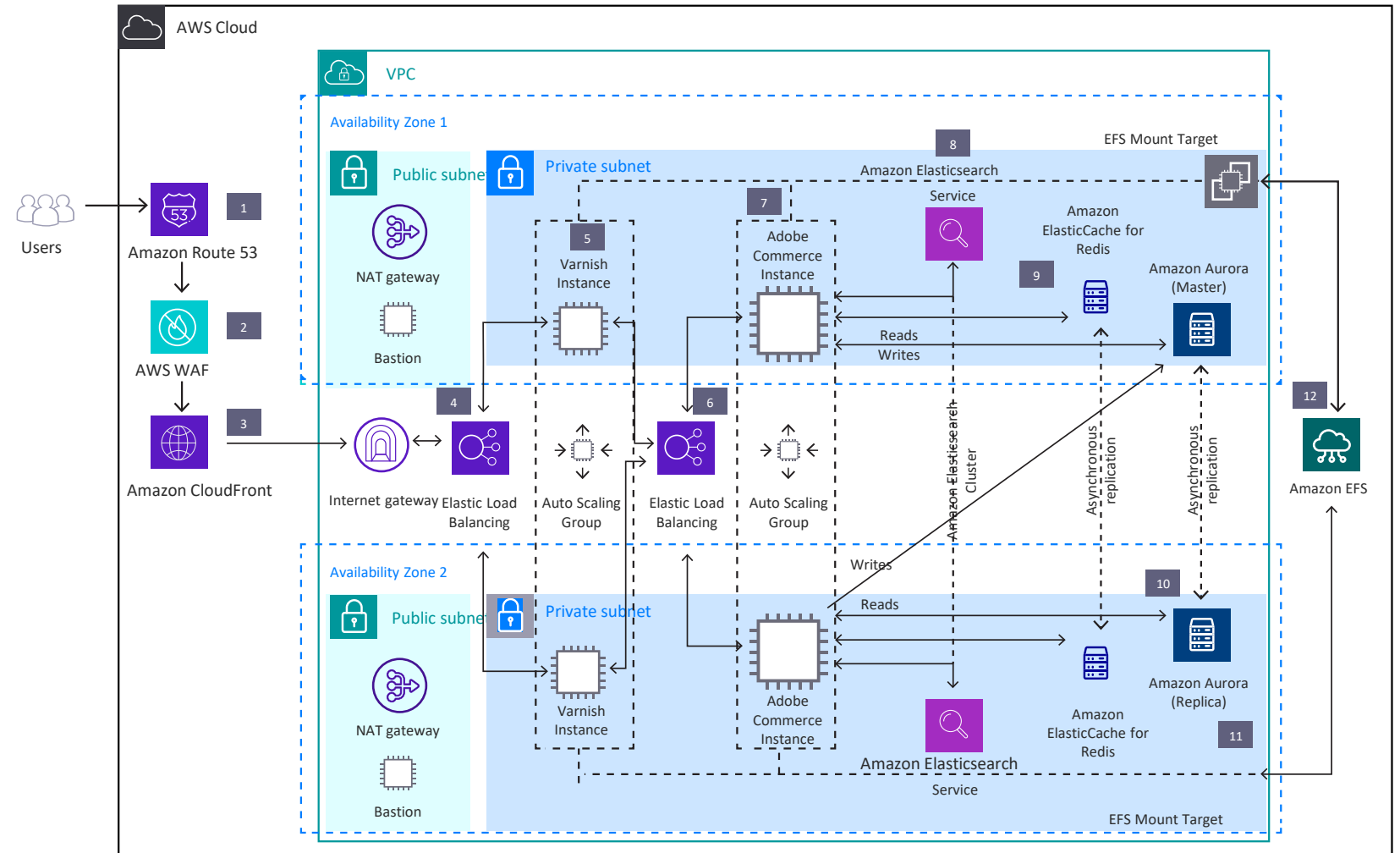
Thus far, we have covered each of the steps necessary to set your ecommerce implementation on Adobe Commerce up for success. Now, we dive further into the actual platform itself, highlighting the infrastructures and environments that are powering Adobe Commerce and the software solutions that make it such a robust platform.

We will also detail a handful of tactical and actionable ways to optimize your Adobe Commerce instance. This includes recommendations and definitions around security, configuration, and performance optimizations. By the time you are done reading this section, you will understand the ins and outs of the platform, as well as easy fixes for allowing it to run at optimum speed.



Adobe Commerce on-premise infrastructure

The motivations for starting a new Adobe Commerce instance or moving an existing on-premise Adobe Commerce instance to the cloud are numerous, but the most common strategic drivers are reducing capital expenditure, decreasing ongoing cost, improving scalability and elasticity, improving time-to-market, and attaining improvements in security and compliance.



The following figure shows the reference architecture for deploying Adobe Commerce on-premise on AWS infrastructure. Other cloud providers like Azure, Google Cloud, and Alibaba Cloud share a similar infrastructure design and homologous services.

Let's dive deeper into the roles and functions of each aspect of the infrastructure.

1. Amazon Route 53 provides DNS configuration.
2. AWS WAF is a web application firewall that protects Adobe Commerce against common web exploits.
3. Amazon CloudFront is a fast content delivery network (CDN) that speeds up the distribution of static and dynamic web content.
4. The first Elastic Load Balancing application load balancer distributes traffic across Varnish instances in an AWS Auto Scaling group in multiple Availability Zones.
5. Varnish Cache is a web application accelerator caching HTTP reverse proxy. The enterprise version, available via AWS marketplace, is recommended as it has better features to support cloud back-ends and cache-purging across dynamic hosts.
6. The second Elastic Load Balancing application load balancer distributes traffic from Varnish Cache across the AWS Auto Scaling group of Adobe Commerce instances in multiple Availability Zones.
7. Install the latest version of Magento open source or Adobe Commerce edition on Amazon EC2 instances. Installation consists of the Adobe Commerce application, Nginx webserver, and PHP. Build the Amazon Machine Image (AMI) to launch new instances in Auto Scaling group.
8. Amazon Elasticsearch Service is a managed elastic search service for Adobe Commerce catalog search.
9. Amazon ElastiCache for Redis provides a caching layer for database.
10. Use Amazon Aurora or Amazon RDS to simplify database administration (including high availability and multi-master configuration).
11. EFS Mount Target facilitates mapping the Amazon Elastic File System (Amazon EFS) to Varnish and Adobe Commerce instances.
12. Use Amazon EFS to access shared config across Varnish and shared media assets across Adobe Commerce instances.

In addition to providing a supporting technology platform for the enablement of DevOps processes on AWS around your Adobe Commerce environment, AWS provides a collection of services that can provide (in the absence of) or augment your existing software configuration management (SCM) solutions. This includes AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, and AWS CodeDeploy, which allows for a managed source control, build, continuous integration/continuous deployment (CI/CD), and deployment services.

The value proposition for migrating Adobe Commerce to AWS is further enhanced by the availability of several services that provide operational insight and agility. What we mean is operational insight into the platform from not only a technical perspective (e.g., requests per hour) but also a business operational perspective (e.g., orders per hour, etc.), particularly when the two sets of data can be married. This provides a near-real-time look into campaign performance, platform operations costs, and a near infinite number of other indicators.

Adobe Commerce setup to AWS can replace specific application dependencies with fully managed alternatives available in the cloud. For example, rather than directly hosting a relational database on EC2 instances, the database for many applications can be easily replaced by Amazon Relational Database Service (Amazon RDS). The benefit to this strategy is that the operating responsibility of undifferentiated components can be offloaded to AWS without requiring significant changes to the core application.

There are several deployment options available for running Adobe Commerce (both Open Source and Adobe Commerce versions) on AWS. The most appropriate choice depends on your requirements for cost, scale, availability, and flexibility, as well as the AWS and Adobe Commerce skills of your organization.



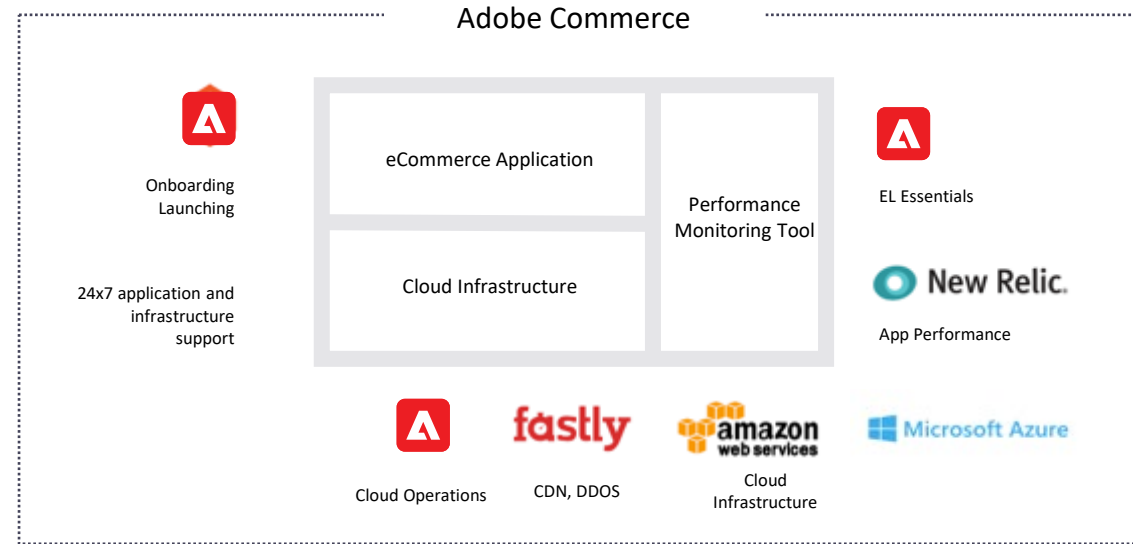
Adobe Commerce Cloud

One of the most popular managed-hosting options for Adobe Commerce on AWS is offered by Adobe Commerce itself. Adobe Commerce Cloud is a fully managed automated hosting platform for the Adobe Commerce software.

Adobe Commerce Cloud is a platform-as-a-service offering that enables rapid deployment of fully customizable, secure, and scalable web storefronts combined with a leading hosting and managed services infrastructure. It offers two plans with different infrastructures. Adobe Commerce Starter is best suited for smaller stores with less complexity and smaller catalogs. Adobe Commerce Pro is built for larger stores with more complexity, larger product catalogs, or traffic that peaks. Adobe Commerce will determine the appropriate architecture with input from partners.

Adobe Commerce is cloud-ready with a fully redundant multi-cloud hosting infrastructure that provides optimized performance, resilience, and elastic scalability. You can efficiently run your commerce platform on Fastly's content delivery, and with New Relic for monitoring and management, you can keep your store environment running smoothly.

Adobe Commerce offers all the benefits of modern cloud computing that are most commonly associated with SaaS solutions: elastic scalability, high resilience and availability, PCI compliance, and global availability and automated patching, while still maintaining flexibility in software customization that our merchants require.

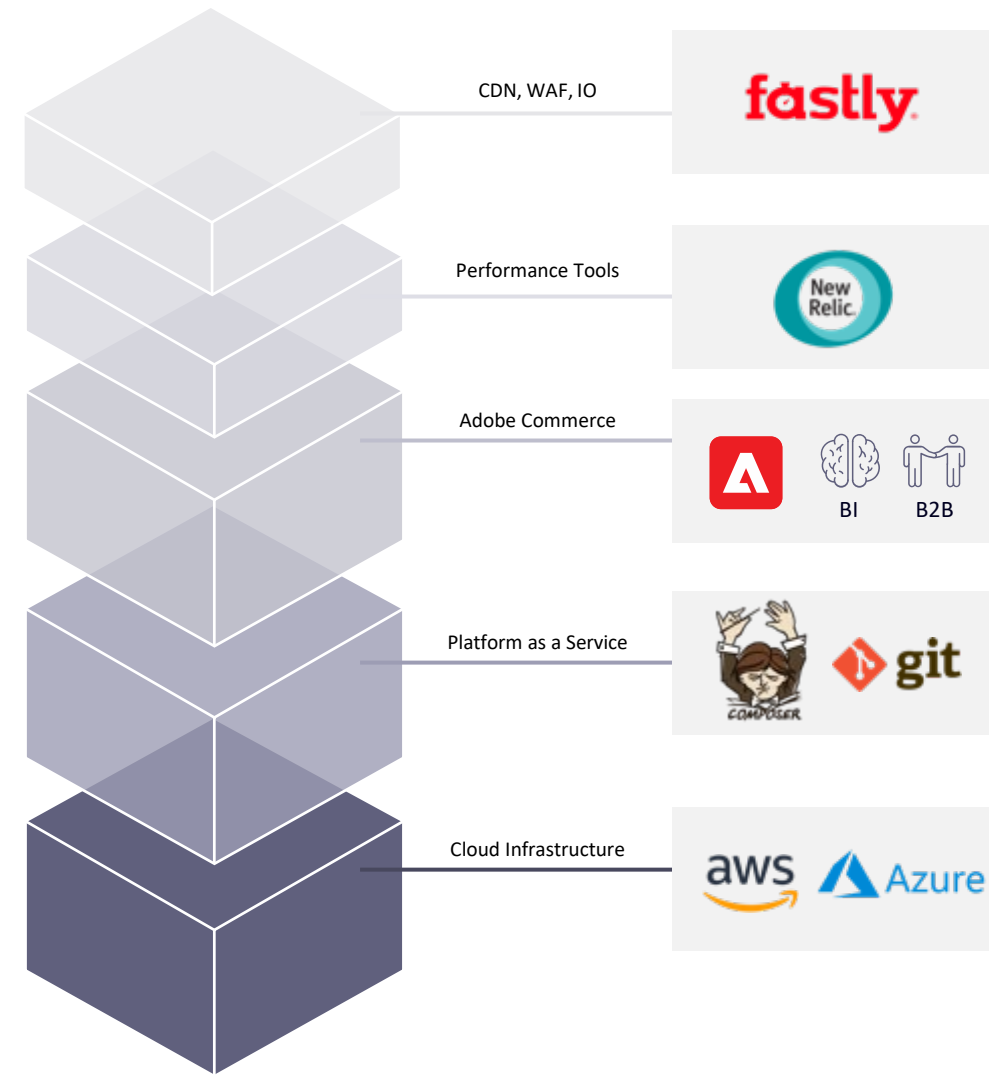


Other benefits of Adobe Commerce include:

- Optimized for Adobe Commerce. Adobe Commerce-developed build scripts and service configuration ensure every instance is correctly tuned and configured for optimal merchant performance.
- Consistent, secure releases. All code deployments are Git-based for consistency and repeatability, with read-only production environments for hardened security.
- Flexibility for partners. A full REST API and a scriptable command line interface ensures ease of integration with external systems and compatibility with existing code management workflows.
- Flexible deployment toolset. Rapidly spin up, merge, clone, and tear down unlimited environments at will for development tasks, QA testing, or production-issue diagnosis.
- Continuous cloud delivery. Move with confidence straight from development to UAT to production, in a continuous manner across code branches and development teams.

Let's also take a look at the software that make Adobe Commerce's benefits a reality.

- Fastly CDN: As customers access your site and stores, the requests hit Fastly to load cached pages faster. Fastly WAF also provides DDoS protection service.
- New Relic gives you a complete view of your applications and operating environment. It allows you to combine key metrics from mobile and browser apps with supporting services, data stores, and hosts so you can optimize performance holistically and ensure the success of every initiative.
- Composer manages dependencies and upgrades in Adobe Commerce Cloud and provides context about the included packages, what the packages do, and how they fit together.
- Git is your code in repositories. It allows local branching, convenient staging areas, and multiple workflows with automatic build and deploy for efficient rapid development and continuous deployment.
- Platform as a Service (PaaS) provides a pre-provisioned infrastructure that includes PHP, MySQL, Redis, RabbitMQ, and Elasticsearch technologies.
- AWS or Azure's cloud hosting powers the underlying Infrastructure as a Service (IaaS), which offers a scalable and secure environment for online sales and retailing.



Adobe Commerce Cloud – AWS regions



NOTE: Only on-premise in China, Russia..

Adobe Commerce Cloud – Azure regions

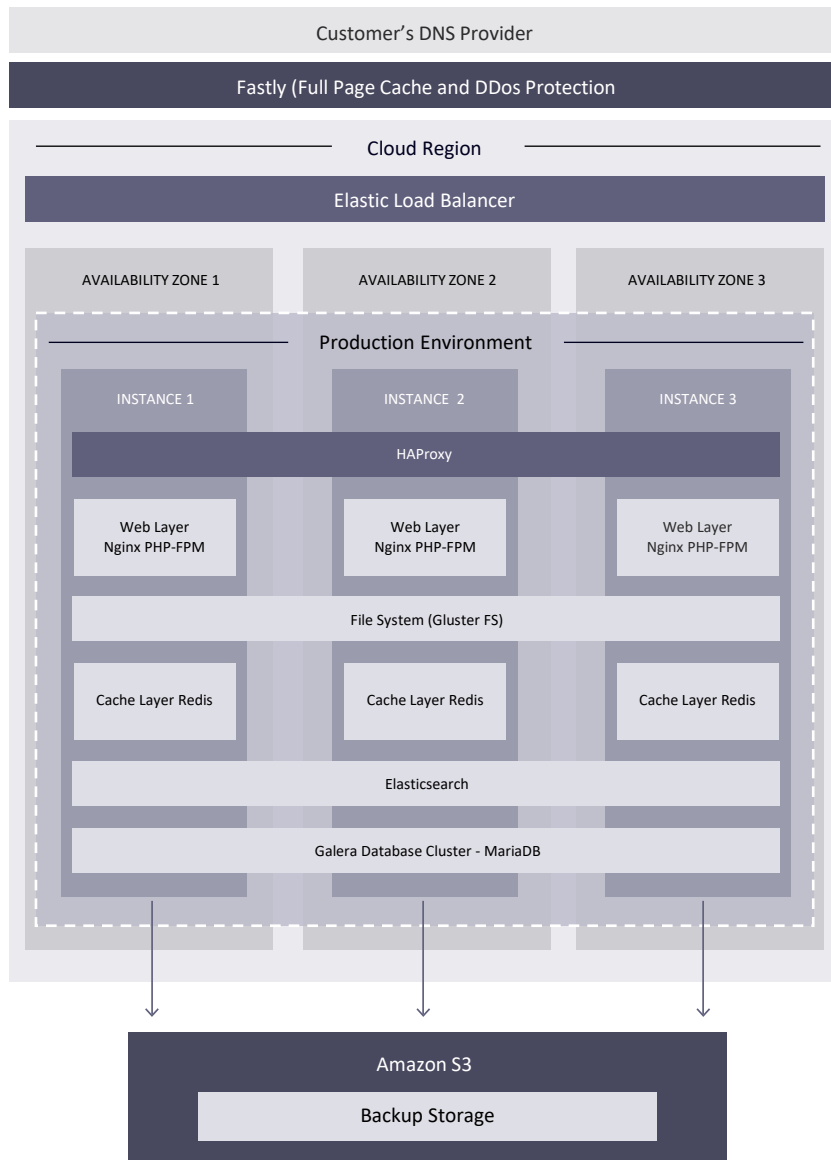


NOTE: Only on-premise in China, Russia..

Adobe Commerce Cloud Technologies Used In Production Environment

As we've mentioned, Adobe Commerce leverages a number of software solutions to support the platform. Specifically, as it pertains to production, we have broken down some of the technical solutions and features included in Adobe Commerce's platform that help make the most of your production environment.





Software solutions

- NGINX: Web server using PHP-FPM. There is one instance with multiple workers.
- GlusterFS: File server for managing all static file deployments and synchronization with four directory mounts: var, pub/media, pub/static, app/etc.
- Redis: One server per VM with only one active and the other two as replicas.
- Elasticsearch: Search for Magento Commerce Cloud 2.2 and newer.
- Galera: Database cluster with one MariaDB MySQL database per node with an auto-increment setting of three for unique IDs across every database.

Features and benefits

- With three dedicated instances in a VPC, there is an elastic load balancer across three separate availability zones or data centers.
- Higher resiliency is provided against events that may cause a single instance to fail (ex: outage of entire AWS availability zone or data center).
- Zero downtime scaling across the entire stack, including web, caching, search, and database, in less than 15 minutes.

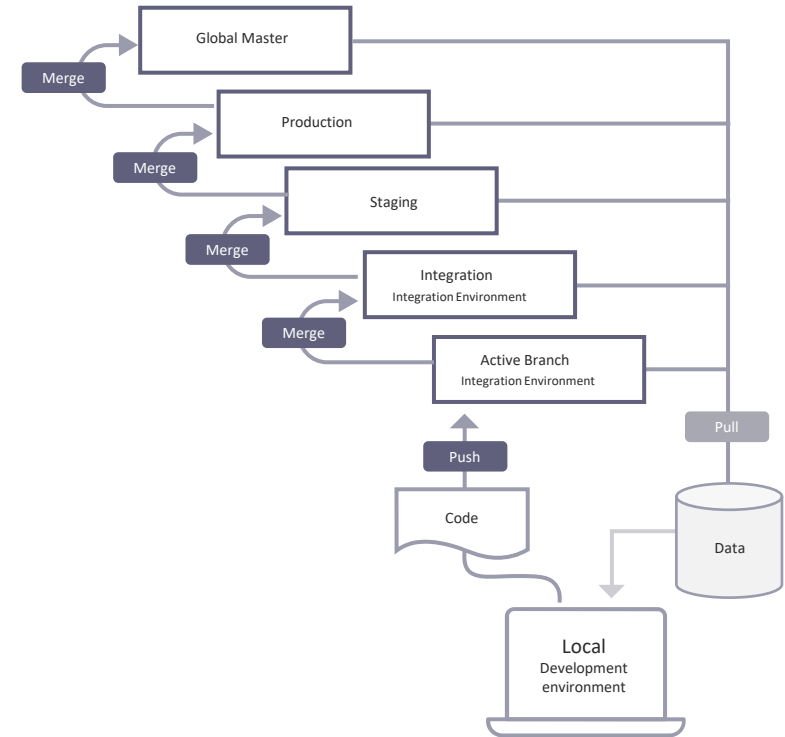
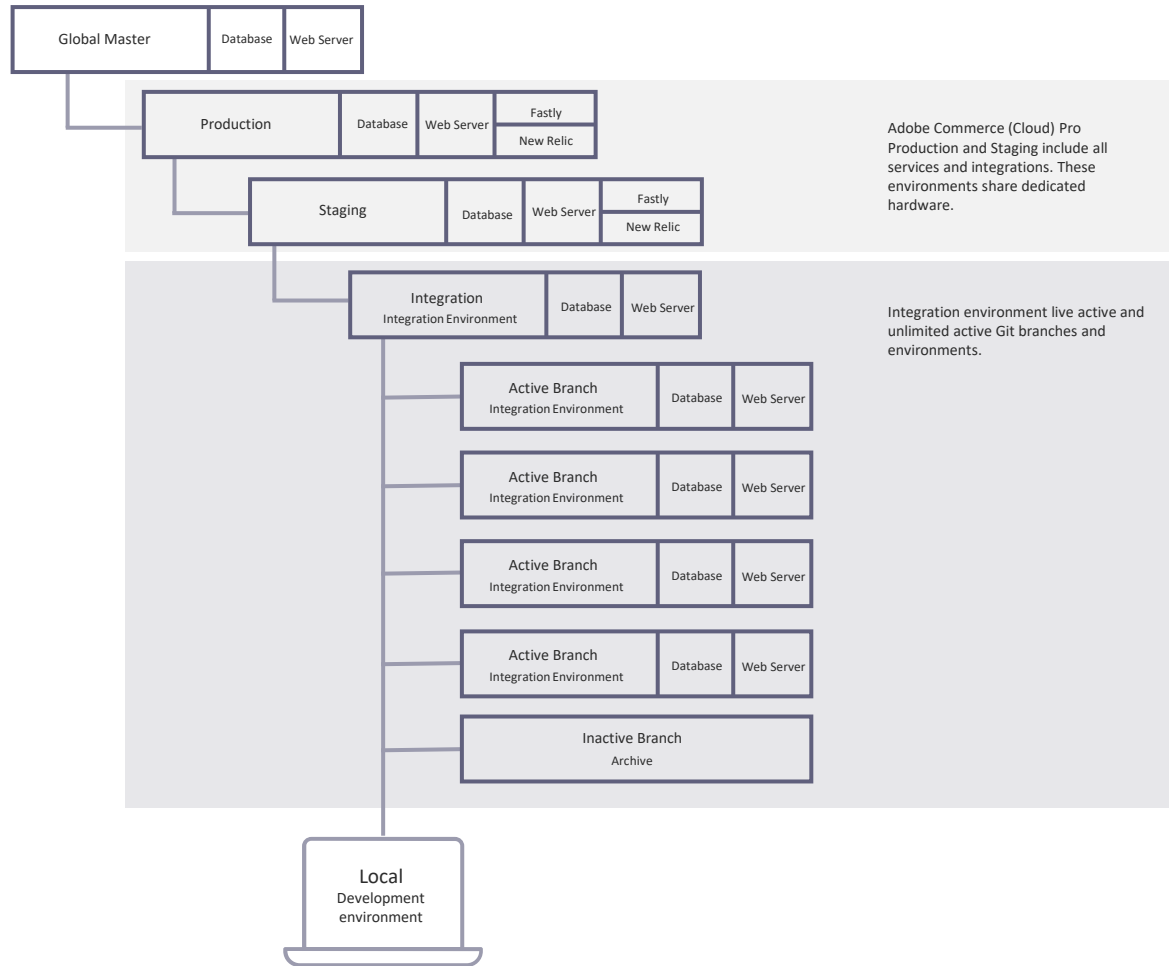
Adobe Commerce Cloud: production, staging, integration

Adobe Commerce Cloud Pro architecture supports environments that you can use to develop, test, and launch your store. Each environment contains a database and a web server. The four environments leveraged by Adobe Commerce are:

- Integration: Provides a single environment branch, and you can create up to four additional environment branches. This allows for a maximum of five active branches deployed to Platform-as-a-Service (PaaS) containers.
- Staging: Provides a single environment branch deployed to dedicated Infrastructure-as-a-Service (IaaS) containers.
- Production: Provides a single environment branch deployed to dedicated Infrastructure-as-a-Service (IaaS) containers.
- Global Master: Provides a master branch deployed to Platform-as-a-Service (PaaS) containers.



Adobe Commerce Cloud: production, staging, integration

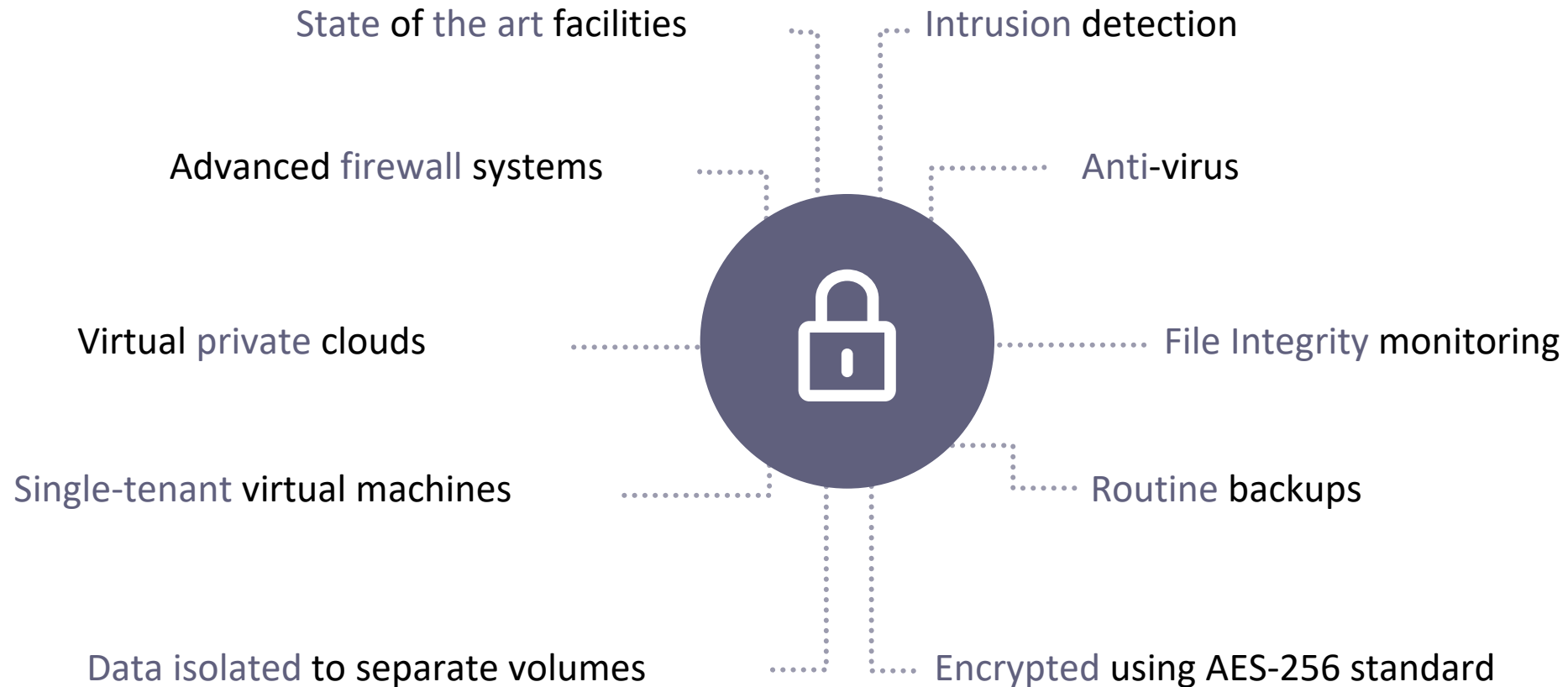


The Integration environment provides a single, base integration branch containing your Adobe Commerce Cloud code deployed to Platform-as-a-Service (PaaS) containers.

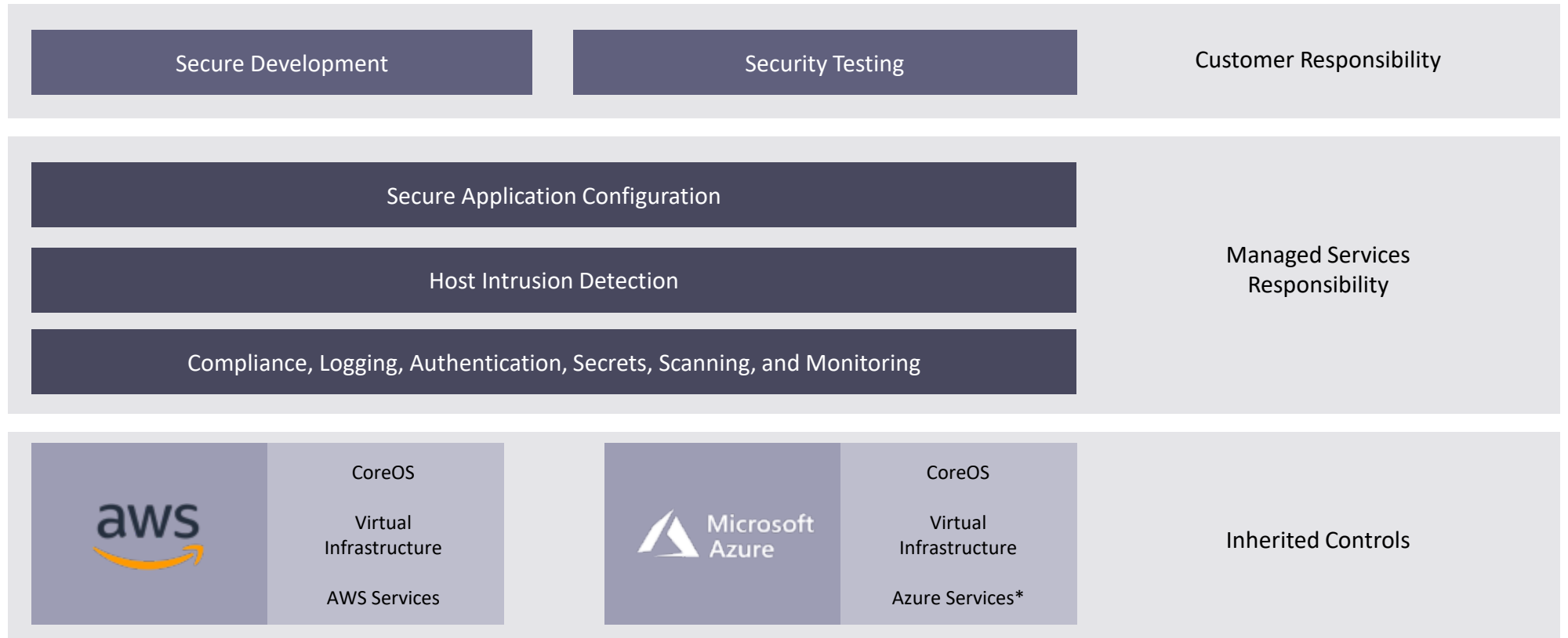
The Adobe Commerce Cloud environments support a flexible, continuous integration process. Begin by cloning the integration branch to your local project folder. Create a new branch, or multiple branches, to develop new features, configure changes, and add extensions. With a developed code branch and the corresponding configuration files, your code changes are ready to merge to the integration branch for more comprehensive testing.



Adobe Commerce Cloud on managed services: secure by default



Adobe Commerce Cloud – shared responsibility security model



Adobe Commerce Pro Cloud relies on a shared responsibility security model. In this model, different parties have different areas of responsibility for maintaining the security of the system. This approach allows for both flexibility and use of best-of-breed cloud technologies.

*Commerce Cloud Pro on Managed Services is currently available with AWS only

Adobe Managed Services Responsibilities

Managed Services is responsible for the security and availability of the Adobe Commerce Pro cloud environment, the core Adobe Commerce Pro application code, and internal commerce systems. This includes, but is not limited to, server-level patching, operating the necessary services to deliver Commerce Pro, vulnerability testing, security event logging and monitoring, incident management, operational monitoring, 24/7 support, and ensuring that the customer infrastructure is available in accordance with SLAs.

Managed Services is also responsible for managing server firewall configurations (Iptables) and perimeter firewall configurations (security groups). Adobe Commerce Pro may also release security updates to the core application on a periodic basis (the application of these patches is the responsibility of the customer). These areas are all covered by the PCI Certification of the Adobe Commerce Pro cloud system.

AWS Responsibilities

Adobe Commerce Pro Managed Services uses Amazon Web Services (AWS) for cloud server infrastructure. AWS is responsible for the security of the network, including routing, switching, and perimeter network security via firewall systems and intrusion detection systems (IDS). AWS is responsible for physical security to the data centers managing the Adobe Commerce Pro cloud environment, as well as environmental security to ensure proper power, cooling, and mechanism controls are in place.

Adobe Commerce Pro uses Amazon's Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3), Elastic Block Store (EBS), Amazon Virtual Private Cloud (VPC), Amazon Elastic Load Balancer (ELB), and Amazon Cloud trail services. Amazon has an extensive compliance program which includes PCI DSS, SOC 2, and ISO 27001 certifications.

Solution Partner/Customer Responsibilities

The customer is primarily responsible for the security of their customized instance of the Adobe Commerce Pro application running in the Adobe Commerce Pro cloud environment. This includes:

- Ensuring a secure configuration and coding of the application and security-monitoring activities including penetration testing and regular vulnerability scans.
- The security of any customization, extensions, other applications, or integrations used in their system.
- The security of their users and the granting of access to their configuration and application.
- The customer controls all code deployments to their non-production environments. This control also comes with the responsibility to apply application security patches to the core Adobe Commerce Pro application, extensions, or any custom code.
- The customer should perform penetration tests of their customized application. These responsibilities can be addressed by technical resources by the customer, implementation partners, or Adobe Commerce Pro professional services.
- Customers are responsible for the PCI requirements of their customized application and their own processes. The customer's PCI compliance builds upon the PCI certifications of Amazon Web Services and Adobe Commerce Pro in order to minimize the areas that must be reviewed.

Adobe Commerce Cloud architecture security

The Adobe Commerce Pro architecture is designed to provide a highly secure environment. Each customer is deployed into their own isolated server environment, separated from other customers. The security details of the production environment are described below.

Consumer's Browser

The bulk of the traffic going in and out of the cloud environment comes from the consumer's browser. Consumer traffic can be secured using HTTPS for all pages on the website (using either a shared SSL certification or the customer's own SSL certificate for an additional fee). Checkout and account pages are always served using HTTPS. The best practice is to serve all pages under HTTPS.

Content Delivery Network (CDN and DDoS Protection)

Fastly provides CDN and DDoS protection. The Fastly CDN helps to isolate direct access to the origin servers. The public DNS only points to the Fastly Network. The Fastly DDOS solution protects against highly disruptive Layer 3 and Layer 4 attacks, as well as more complex Layer 7 attacks. Layer 7 attacks can be blocked using custom rules based on the entire HTTP/HTTPS requests and based on client and request criteria, including headers, cookies, request path, and client IP, or indicators like geolocation.

Web Application Firewall (WAF)

The Fastly Web Application Firewall (WAF) is used to provide additional protection. Fastly's cloud-based WAF uses third-party rules from commercial and open-source sources such as the OWASP Core Ruleset. In addition, Adobe Commerce Pro-specific rules are employed. Customers are protected from key application-layer attacks, including injection attacks and malicious inputs, cross site scripting, data exfiltration, HTTP protocol violations, and other OWASP Top 10 threats.

The WAF rules are updated by Adobe Commerce Pro should new vulnerabilities be detected allowing Managed Services to “virtually patch” security issues in advance of software patches. The Fastly WAF does not provide rate-limiting or bot-detection services. If desired, customers can license a third-party bot-detection service compatible with Fastly.

Virtual Private Cloud

The Adobe Commerce Pro production environment is configured as a Virtual Private Cloud (VPC) so that production servers are isolated and have limited ability to connect into and out of the cloud environment. Only secure connections to the cloud servers are allowed. Secure protocols like SFTP or rsync can be used for file transfers.

Customers can use SSH tunnels to secure communications with the application. Access to the AWS Private Link can be provided for an additional fee. All connections to these servers are controlled using AWS Security Groups, a virtual firewall that limits connections to the environment. Customers' technical resources may access these servers using SSH.

Encryption

Amazon Elastic Block Store (EBS) is used for storage. All EBS volumes are encrypted using the AES-256 algorithm. This means that the data will be encrypted at rest. The system also encrypts data in transit between the CDN and the origin, and between the origin servers. Customer passwords are stored as hashes. Sensitive credentials, including those for the payment gateway, are encrypted using the SHA-256 algorithm.

The Adobe Commerce Pro application does not support column- or row-level encryption or encryption when the data is not at rest or not in transit between servers. The customer can manage encryption keys from within the application. Keys used by the system are stored in AWS Key Management System and must be managed by Managed Services in order to provide parts of the service.

Penetration Testing

Managed Services conducts regular penetration tests of the Adobe Commerce Pro cloud system with the out-of-the-box application. Customers are responsible for any penetration testing of their customized application.

Payment Gateway

Adobe Commerce Pro requires payment gateway integrations where credit card data is passed directly from the consumer's browser to the payment gateway. The card data is never available on any of the Adobe Commerce Pro Managed Services environments. Actions on the transactions by the ecommerce application are completed using a reference to the transaction from the gateway.

Adobe Commerce Pro Application

Adobe Commerce Pro regularly tests the core application code for security vulnerabilities. Patches for defects and security issues are provided to customers. The Product Security Team validates Adobe Commerce Pro products following OWASP application security guidelines. Several security vulnerability assessment tools and external vendors are used to test and verify compliance. Security tools include Veracode Static and Dynamic Scanning, RIPS source code scanning, Trustwave's and Alert Logic's vulnerability scanning services, Burp Suite Pro, OWASP ZAP, and SqlMap. The full code base is scanned with these tools on a bi-weekly basis. Customers are notified of security patches via direct emails, notifications in the application, and in the Security Center (<https://magento.com/security>).

Customers must ensure that these patches are applied to their customized application within 30 days of release, according to the PCI guidelines. Adobe Commerce Pro also provides a Security Scan tool that enables merchants to regularly monitor their sites and receive updates about known security risks, malware, and unauthorized access. Security Scan is a free service and can be run on any version of Adobe Commerce Pro. To encourage security researchers to identify and report vulnerabilities, Adobe Commerce Pro has a [bug-bounty program](#) in addition to internal testing by Adobe Commerce Pro. Further, the customer is provided the full source code of the application for their own review if desired.

Read-Only File System

All the executable code is deployed into a read-only file system image, which dramatically reduces the surfaces that are available for attack. The deployment process creates a Squash-FS image. This approach dramatically reduces opportunities to inject PHP or JavaScript code into the system or modify the Adobe Commerce Pro application files.

Remote Deployment

The only way to get executable code into the Managed Services production environment is to run it through a provisioning process. This involves pushing source code from your source repository into a remote repository that initiates a deployment process. Access to that deployment target is controlled, so you have complete control over who can access the deployment target. All deployments of application code to the non-production environment are controlled by the customer.

Logging

All AWS activities are logged in AWS CloudTrail. Linux, Application Server, and Database logs are stored on the production servers and stored in backups. All source code changes are recorded in a Git repository. Deployment history is available in the Adobe Commerce Pro user interface. All support access is logged and support sessions are recorded.

Sensitive Data

Sensitive data can cover either personal information from consumers or confidential data from Managed Services customers. Protection of sensitive customer and consumer data is a critical obligation for Adobe Commerce Pro Managed Services. Both Managed Services and our customers have legal obligations around personally identifiable information. In addition to the security features of the architecture, there are other controls to limit the distribution and access to sensitive data.

Customers own their data and have control over where that data will be located. The customer specifies the location where their production and development instances reside. They also specify which location will be used for the Magento Business Intelligence (MBI) environment in conjunction with Commerce, and if that MBI application has access to personally identifiable information or not. Production instances can be located in most AWS regions, while development and MBI environments can be found in either the United States or in the European Union at this time.

Sensitive data may pass through the Fastly CDN server network but is not stored in the Fastly network. All partners included in the Adobe Commerce Pro Managed Services offering have contractual obligations to ensure the protection of sensitive data. Managed Services will not move sensitive customer or consumer data from the locations specified by the customer.

As part of providing the services included in the Adobe Commerce Pro Managed Services offering, Managed Services staff requires access to production systems that contain sensitive data. These employees are trained on their obligations to protect sensitive customer and consumer data. Access to these systems is limited to employees that require access. These employees only have access for the time needed to deliver these services.

General Data Protection Regulation (GDPR) Overview

GDPR is a legal framework that sets guidelines for the collection and processing of personal information for individuals in the European Union (EU). These regulations apply regardless of where the site is based, and EU visitors have access to it.

Essentially, visitors must be notified of the data the site collects from them and explicitly consent to information gathering. Sites must notify visitors if personal data held by the site is breached.

The merchant or company operating the site must also have a dedicated Data Protection Officer who oversees the site's data security, and this individual (or website management team) should be available for contact should a visitor request that their presence / data on the site is erased.

GDPR also calls for any personally identifiable information (names, race, date of birth, etc.) collected to be either anonymized or pseudonymized.

Note: This page is simply meant as a general overview for what to consider for GDPR. For more information, please consult with your legal counsel or refer to the [official text](#).

Backups

Backups are performed every hour for the last 24 hours of operation. After the 24-hour period, backups are retained on the following schedule below, using the AWS EBS Snapshot service. This creates an independent backup on redundant storage. Because the EBS volumes are encrypted, the backups are also encrypted. Additionally, Managed Services performs on-demand backups on customer-requested basis.

Your Adobe Commerce Pro Managed Services backup and recovery approach uses a high-availability architecture combined with full-system backups. Each project is replicated—all data, code, and assets—across three separate AWS availability zones; each zone with a separate data center.

TIME PERIOD	BACKUP RETENTION POLICY
Days 1 to 3	Each Backup
Days 4 to 6	One Backup per day
Weeks 2 to 6	One Backup per week
Weeks 8 to 12	One bi-weekly backup
Weeks 12 to 22	One backup per month



Performance optimization

Performance is a big topic. When users experience a slow feedback from any touchpoint, it is affected by many factors. To optimize, the follow steps of performance tuning are recommended:

- Assess the problem
- Measure performance
- Identify part of the system critical for performance improvement
- Modify part of system to remove the bottleneck
- Measure the performance after modification
- If better, adopt it or revert

Typical Performance Issues

The impact of slow experience usually defined by two indicators, and each factor can be caused for tons of reasons.

High TTFB (Time to First Byte) is usually regarded as an indicator that defines the server's response speed. The time not only comes from source code execution for handling the request, but it can also impact by follow factors:

- DNS lookup
- Slow queries from DB layer
- CPU time from each application layer
- Memory limitation
- I/O wait may affect from file read and write, connect service via socket.
- Software settings (Nginx, PHP, MySQL, Redis, Varnish, etc.)
- Network bandwidth
- Bad caching
- Bad codes
- Bad integration approach
- Dependency of slow third-party service response
- Architecture without scalability

Slow-loading resources is usually regarded as an indicator that defines the static resource (CSS, JavaScript, images, videos, third-party ajax call response, etc.).

There are also key factors driving scale in commerce, which will also impact the overall performance.

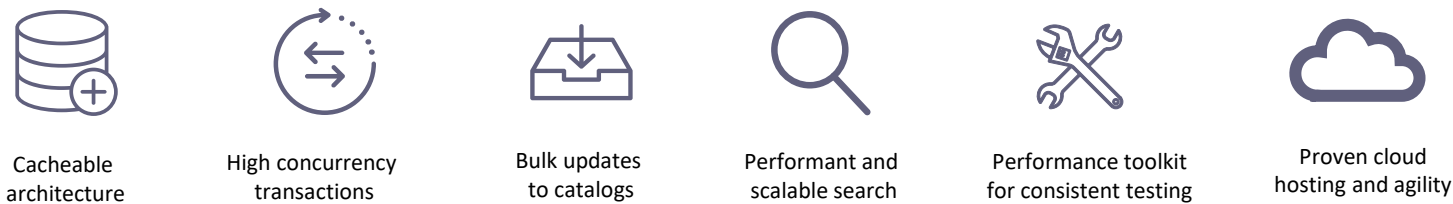
Typical Performance Issues

- Complex and large product catalog
- Large numbers of admins
- Global storefronts

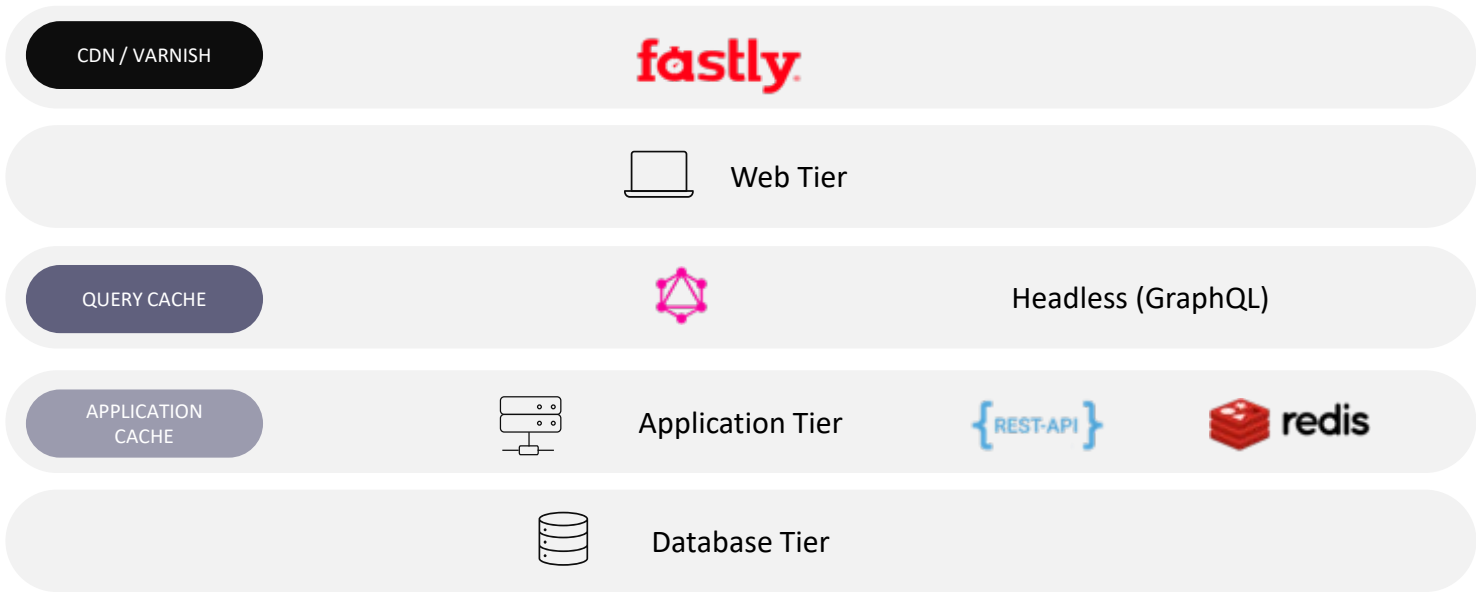
Shopping Experiences During Peak Time

- High-variable traffic
- Expanding touchpoints
- High-volume transactions

Adobe Commerce can scale with your business through its capabilities:



For layered and cacheable architectures built for scale, you can use this graph as a reference.



Adobe Commerce Cloud Pro without additional customizations and CDN cache has the following benchmark figures:

200M

Effective SKUs

300ms

Average API
Response Time

1,700

Orders Per Hour

<2s

Average Storefront
Response Times

500k

Hourly Page Views





Adobe Commerce performance optimization review

Even as performance optimization can come from many aspects, there are some general recommendations that should be considered for most scenarios. This includes configuration optimization for infrastructure elements, Adobe Commerce back-end configuration, architecture scalability planning, etc.

Infrastructure

1. Reduce DNS lookups

DNS lookup is the process of finding which IP the domain name belongs to. The browser must wait until the DNS lookup is complete before it can download anything for each request. So, reducing DNS lookups is important to improve overall page load times

2. CDN to optimize asset downloading performance

Adobe Commerce Cloud uses Fastly. Those using On-Premise Adobe Commerce should also consider adding a CDN layer.

3. Optimize web latency

The location of the data center choice will affect the web latency for target front-end user.

4. Network bandwidth

Sufficient network bandwidth is one of the key requirements for data exchange between web nodes, database(s), caching/session servers, and other services. Because Magento 2 effectively leverages caching for high performance, your system can actively exchange data with caching servers like Redis. If Redis is located on a remote server, you must provide a sufficient network channel between web nodes and the caching server to prevent bottlenecks on read/write operations.

5. Operating system kernel optimized for web application server.

Operating system configurations and optimizations are similar for Adobe Commerce as compared to other high-load web applications. As the number of concurrent connections handled by the server increases, the number of available sockets can become fully allocated.

6. CPU of web nodes

One CPU core can serve around 2-4 Adobe Commerce requests without cache effectively. To determine how many web nodes/cores needed to process all incoming requests without putting them into queue, use the equation:

$$N[\text{Cores}] = (N[\text{Expected Requests}] / 2) + N[\text{Expected Cron Processes}]$$

7. PHP / PHP-FPM Settings Optimization

Process Manager Optimization depends on the performance test results for different projects.

- **ByteCode:**
To get maximum speed out of Magento 2 on PHP 7, you must activate the OpCache module and properly configure it.
- **APCU:**
We recommend enabling the PHP APCu extension and supporting configuring composer to optimize for maximum performance. This extension caches file locations for opened files, which increases performance for Adobe Commerce server calls, including pages, Ajax calls, and endpoints.
- **Realpath_cache configuration:**
Optimizing realpath_cache allows PHP processes to cache paths to files instead of looking them up each time a page loads.

8. Nginx Settings Optimization

Only slight reconfiguration is needed to use Nginx as a web server application for better performance and sample configuration from Adobe Commerce.

Setup PHP-FPM with TCP properly
Enable HTTP/2 and Gzip
Optimize worker connections

9. MySQL (MariaDB/Percona) Settings Optimization

This document does not provide in-depth MySQL tuning instructions because each store and environment is different, but we can make some general recommendations.

The Magento 2 database (as well as any other database) is sensitive to the amount of the memory available for storing data and indexes. To effectively leverage MySQL data indexation, the amount of memory available should be, at minimum, close to half the size of the data stored in the database.

Optimize the `innodb_buffer_pool_instances` setting to avoid issues with multiple threads attempting to access the same instance. The value of the `max_connections` parameter should correlate with the total number of PHP threads configured in the application server. The best value for `innodb-thread-concurrency` should be calculated by the formula: `innodb-thread-concurrency = 2 * (NumCPUs + NumDisks)`

10. Redis with optimize settings for data cache and session

Session cache is a good candidate to configure for a separate instance of Redis. Memory configuration for this cache type should consider the site's cart-abandonment strategy and how long a session should expect to remain in the cache.

Redis should have enough memory allocated to hold all other caches in memory for optimal performance. Block cache will be the key factor in determining the amount of memory to configure. Block cache grows relative to number of pages on a site (number of SKU x number of store views).

11. Varnish for FPC (Full page cache)

Adobe Commerce highly recommends using Varnish as the full-page cache server for your store. The PageCache module is still present in the codebase, but it should be used for development purposes only.

Install Varnish on a separate server in front of the web tier. It should accept all incoming requests and provide cached page copies. To allow Varnish to work effectively with secured pages, an SSL termination proxy can be placed in front of Varnish. Nginx can be used for this purpose.

While Varnish full-page cache memory invalidation is effective, we recommend allocating enough memory to Varnish to hold your most popular pages in memory.

12. Elasticsearch to be the catalog search engine

Elasticsearch is required as of version 2.4, but it's also best practice to enable it for versions before 2.4.

13. Message queues to enable asynchronized process

The Message Queue Framework (MQF) is a system that allows a module to publish message to queues. It also defines the consumers that will receive the messages asynchronously. Adobe Commerce supports RabbitMQ as the messaging broker, which provides a scalable platform for sending and receiving messages.

14. Performance Testing and Keep Monitoring

Performance testing before each production release is always recommend to get an estimation of the capability of entire ecommerce platform. Keep monitoring after launch and have a scalability and backup plan how to handle peak time.

NOTE: Adobe Commerce Cloud already applies all of the above infrastructure and architecture optimizations except for the DNS lookup, as it's out of scope.

Apart from those common optimization items, there are also approaches to enhance the performance for specific business modes and scales. This document does not provide in-depth tuning instructions for all of them, because each scenario is different, but we can provide a few high-level options for your reference.



1. Headless architecture

We have a separate section dedicated to detailing what headless is and different options. In summary, it separates the storefront layers from the platform itself. It is still the same back end, but Adobe Commerce no longer processes request directly, and instead is only supports custom storefront with API powered by GraphQL.

2. Keep Adobe Commerce updated

Adobe Commerce always has better performance when running the newest version. Even if it's not possible to keep Adobe Commerce up-to-date after each new version is released, it's still recommended to upgrade when Adobe Commerce introduces significant performance optimizations. For example, in 2020, Adobe released an optimization to the Redis layer, fixing a lot of inefficiencies, connection issues, and unnecessary data transfer between Redis and Adobe Commerce. Overall performance between 2.3 and 2.4 is night and day and we saw significant improvements in cart, checkout, concurrent users, etc., just because of the Redis optimization.

3. Optimize Data Model

A lot of problems originate from data—bad data models, data that is not structured properly, or data that is missing an index. It will look

fine if you're testing a few connections, but seen in production when the real traffic hits—and this is where slowness comes in. It's very important that from the beginning of the project for the systems integrator to know how to design a data model (especially for product attributes), avoid adding unnecessary attributes, and keep mandatory attributes that affect to business logic (pricing, stock availability, search).

For those attributes not affecting business logic but still need to be present on the storefront, combine them into a few attributes (e.g., json format). To optimize platform performance, if business logic is not required on the storefront from data or attributes taken from a PIM or an ERP, there is no need to add that attribute into Adobe Commerce.

4. Design Adobe Commerce with Scalability

This is important for businesses running campaigns and facing peak times often. For architecture and application design to be easy to scale, this can increase resources during peak times and reduce them after it.

Launch + Cut-Over Processes

ADOBE COMMERCE IMPLEMENTATION PLAYBOOK

07

Checking every box

By this point in your ecommerce journey, you teams and organization have dedicated months to organizing, planning, and building a solution that fits your brand's needs and will grow your business. Now that the point of launch is finally here doesn't mean the job is now done.

From the final pre-launch checkpoints to the post-launch steps that will ensure the site continues to maintain the level of effectiveness from day one on, we have highlighted all of the tactics and actions that should take place in the four key stages from testing to post-launch. Save these checklists, as they are your guide to a smooth launch and the payoff to all of those months of hard work.





Pre-launch readiness

When you have completed deployment and testing in the staging environments, you can begin site-launch preparation. Staging is a near-production environment running on similar hardware, configurations, architecture, and services. It can reduce your downtime and make your extension, service, custom configurations, and merchant user-acceptance testing vital components to releasing your sites and stores.

The pre-launch checklist is required to verify prior to launch state, which includes the following major verifications:

- Code freeze for deployment
- Ensure that the downtime was communicated in advance by at least one day for maintenance release and one week for the first launch
- Deployment scripts are set up/configured completely for Production/Staging/Integration environments
- Databases are all set up and identical between Staging and Production environments
- SSL (TLS) certificates are validated for Staging/Production environments
- Mail services are well-configured and functioning for transactional emails
- CDN is configured for Staging/Production environments
- Set up security scan for Staging/Production environments
 - Adobe Commerce security scan
- Perform performance assessment by
 - JMeter
 - Siege
 - Webpage Test
 - Google page speed
- Validate all third-party integrations that will be functioning in application (OMS, CRM, etc.)
- Enable performance-monitor tool (New Relics)
- Data migration activities in rehearsal (if any)



The main differences between On-Premise Adobe Commerce and Adobe Commerce Cloud are the deployment scripts and tools, as well as the setup for SSL, Mail service, and CDN. However, the process is still the same.

For SSL(TLS) certificate, Adobe Commerce Cloud provides a Fastly wildcard certificate. To start using it, you need to pass the validation: add the Fastly TXT record to apex domain name within your DNS settings. The Fastly TXT record can be found in the on-boarding spreadsheet, otherwise you need to submit support ticket to obtain it.

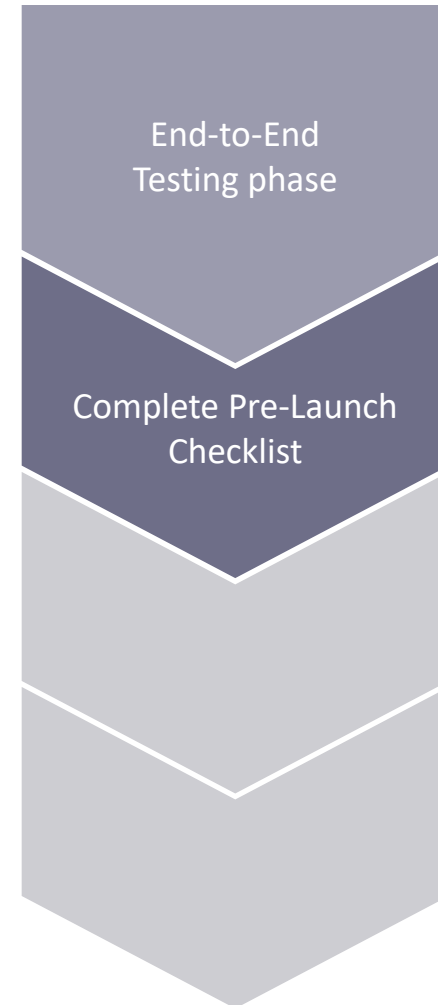
Replace this text with your questions/comments here. If you use your own SSL(TLS) certificate instead of a Fastly wildcard certificate, submit a support ticket with your certificate attached to the setup.

Adobe Commerce (Cloud) provides SendGrid Mail functionality for your Adobe Commerce transactional emails. For Pro, you need to add SendGrid records to your DNS settings. SendGrid records can be found in the on-boarding spreadsheet, otherwise SI or merchant should submit support tickets to obtain them. To start, you don't need to make any changes to your DNS; SendGrid is pre-configured for you.

Complete pre-launch checklist

The complete pre-launch checklist shows all major activities whose completeness is necessary for moving to the launch state.

- Go-live risk mitigation plans updated
- Correct domain names provided
- Outgoing emails have been tested
- SSL certificate is provisioned and configured
- All-important configuration of Adobe Commerce application is updated correctly
- Base URLs and Base Admin URL are correctly set to final hostname
- Admin passwords are changed
- All users with access to application that no longer require access are removed
- Payment configuration for production environment (for some, payment is using sandbox mode for testing)
- Test data (customer, wishlist, reviews, orders, and related data) from Production database is cleared



Launch steps

After testing and completing the pre-launch checklist, we can start the final steps to launch at the time to cutover. These steps include entering site launch (go live) tickets, cutting over access, and finally, testing your store(s) when live.

Adobe Commerce support staff will work with you through the process, checking the status and helping to address any questions or issues that occur. All issues should be tracked with tickets to best capture what happened and how it was resolved. When you begin deploying continuous iterations of updates to your launched store, you may have similar issues occur again. These tickets can help pinpoint the issues and help adjust your deployment tasks.

- Configure application to Base URL
 - Switch DNS to the new site
 - Access your DNS service
 - Update your A and CNAME records for your domains and hostnames
 - Wait for the TTL time to pass and access your store
- Completely test in Production
 - Verifying all functions of the website
 - Verifying CDN cache
 - Verifying all integrated third-party services
 - Verifying all third-party systems
- Contact Adobe Commerce hotline in case any issues are blocking the go-live



Post-launch steps

Once the website is live, these activities would be performed as soon as possible to ensure the site was launched properly.

- Enable up-time monitoring tool (New Relics), activate checks, and monitor the site to ensure all services and access are in the green
- Enable Adobe Commerce security scan periodically
- Tag the cluster as live and create a support ticket to activate High SLA monitoring
- The CSE (Customer Success Engineer) and TAM (Technical Account Manager) will perform the following tasks as soon as cutover is completed:
 - Tag the cluster as High SLA for Adobe Commerce Cloud client and create a support ticket to activate it
 - Activate the Pingdom checks for domain names
 - Review monitoring state and ensure all items are in green
 - Keep the stakeholders informed of the warranty duration and parameters via email on go-live day



Ongoing Support + Maintenance

ADOBE COMMERCE IMPLEMENTATION PLAYBOOK

08

Rethinking support

Consumers already have more options than ever before at their fingertips. With no shortage of brands vying for attention, you can't give consumers any reason to look at your competitors. As we've seen, loyalty and patience with consumers is thin. It doesn't take much of them to give up on your brand and having a poor ecommerce experience is an easy way for them to give up.

This leads to two complementary points. The first is that the launch of a new ecommerce site does not mean you then get to move on. The rate of change in marketing and with consumers' needs is too great that brands must constantly evolve to keep up. Which brings us to point two. If your ecommerce support is only there to fix something when it breaks, then it will be impossible to meet consumers' rising expectations. In short, ecommerce support shouldn't just keep your site working, it should move your brand forward. This section will help you begin to move your brand forward after the launch of your site.



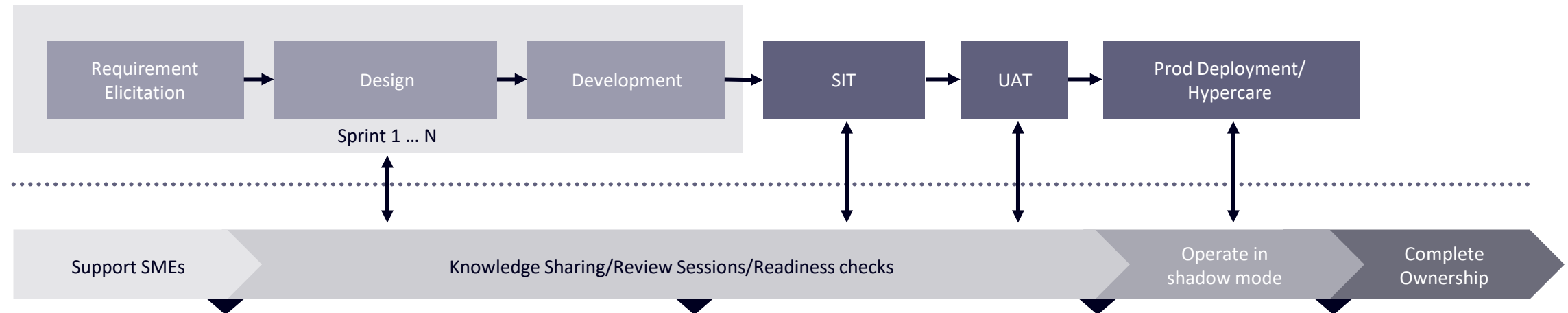


The transition phase

Setting up production support during the transition phase of a project is one of the most critical success factors for a commerce enterprise. Once the implementation is completed and the site goes live, the production support team is required to be ready and equipped to take over support activities. The usual practice is to ramp down the development team during transition phase and build a smaller team in place for support.

Knowledge transfer occurs over the course of the entire project, and a successful transition approach occurs in parallel with delivery. In addition, user guides and a technology wiki are important tools that enable the team through workshops throughout the project phases.

A look at the phases and activities that would be included in a successful transition outcome



Support Team

- Business to Run objectives
- Functional and Technical scope
- Impact on current set up—application, environment, processes
- Revised AOD (application overview document)
- QA – Test scenario review
- Define Training plans
- Evaluate Run readiness
- Work along with Dev Warranty team for ticket closures
- Updated SOPs/alerts
- Warranty Plan Exit criteria review/sign off
- Reverse shadow for final week
- Ownership of new application

It is important to put together a transition checklist that helps project managers complete the tasks that are required to successfully set up the post-production support team. This transition should be part of the overall project plan and the tasks have to be included in the schedule.

Types of maintenance and support models

There is not just one way to handle the support of your ecommerce site. We have outlined four options, depending on the needs and goals of your brand.

The option of managed services with an SLA commitment and fixed maintenance capacity is an outcome-driven model to manage, maintain, and enhance digital commerce ecosystems with expected service levels and to produce strong, quality, and measurable outcomes for digital business and operations with fixed capacity and cost.

01 Fixed capacity production support and enhancements

Fixed capacity refers a restricted number of committed hours for resources as contracted. For simple commerce sites, this number can typically vary from 40 to 160 hours per month. The nature of work and type of resources depend on the scope of the support needed. This is suited for smaller organizations with relatively simple digital ecosystems, and also in specific engagements where the brand already has a sizable technology team.

03 SLA support for critical incidents with dedicated delivery capacity

- Optional 24*7 support with SLA for severity 1 incidents
- Fixed delivery resources and unlimited operational support
- Additional delivery resources to follow agreed change management process
- Covers all roles (Dev/QA/SA/PM/RM) and operational support

02 SLA Support with hotfix

- Optional 24*7, severity 1 support
- SLA for all support incidents
- Monthly allocation of resources for support/delivery effort
- Additional resources to follow agreed change management process

04 Managed service support with a fixed maintenance capacity

- SLA for all support incidents, including 24*7 response time
- Fixed delivery resources and unlimited operational support to meet committed SLA
- Complete DevOps for HotFix support
- Additional delivery resources to follow agreed change management process
- Covers all roles (Dev/QA/SA/PM/RM) and operational support

Typical service levels are defined as operations and break-fix, enhancements (Also called Level 3 or L3 services) and optional services (Optional L3 services). Operations and break-fix activities can be further divided into L1 (Level 1: monitoring and incident management) and L2 (Level 2: problem management and continuous service improvement). Scope of these activities are provided below.

01

Operations and Break-Fix

- Event management
- Incident management
- Service requests
- Priority notification and reporting
- Problem management
- Change management
- Peak support

02

Enhancements

- Backlog grooming + prioritization
- Sprint + release planning
- Design + development
- Functional sprint testing
- Supporting performance and security testing
- Deployment planning

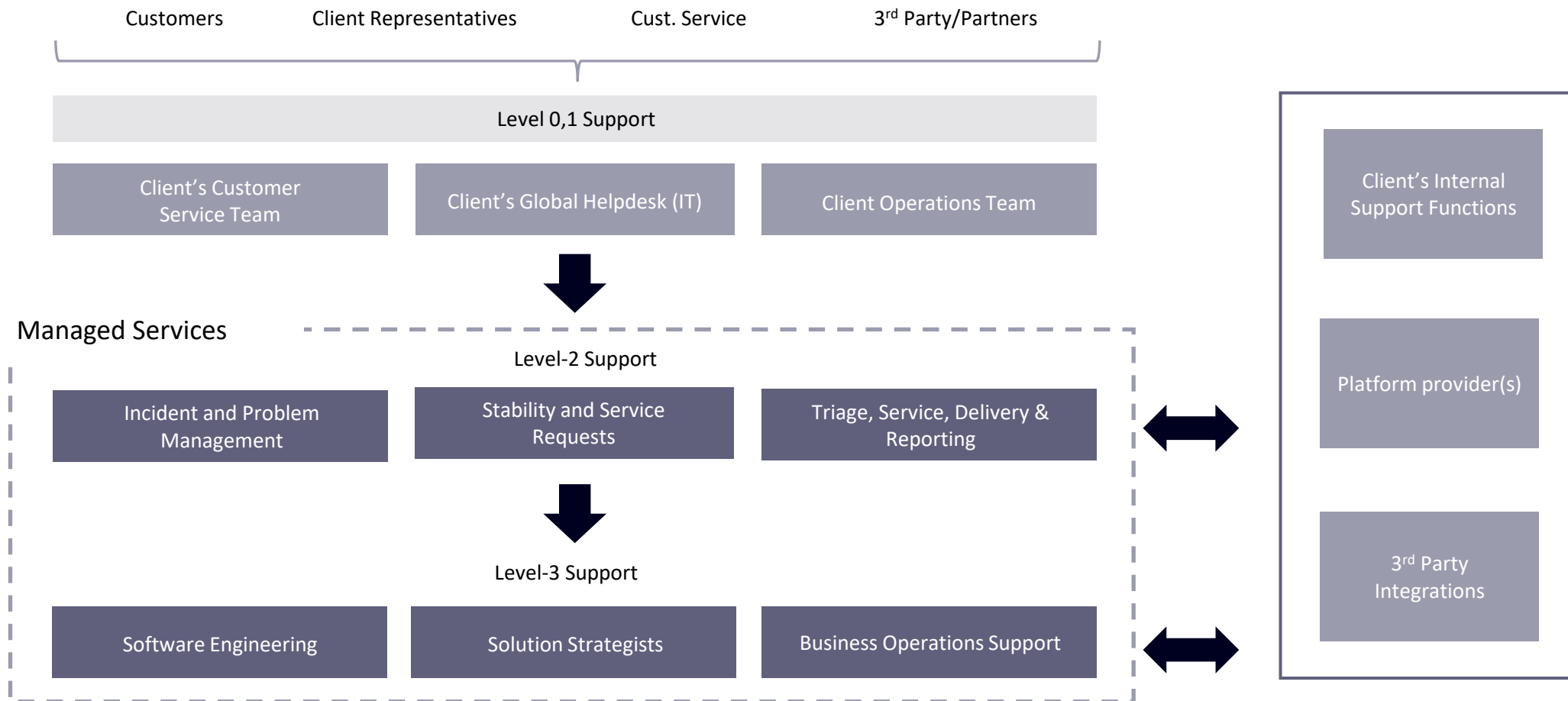
03

Optional Services

- Architecture Audit and recommendation
- Performance and Security Testing
- Business Operations support
- Content production
- Merchandizing Operations
- Strategic consulting
- Adobe Commerce updates

Delivery and service management is also an integral part of support and enhancement function. Here are the activities typically included in the scope under this function:

- Delivery and service oversight, floor and shift management
- Incident and problem triaging
- Stakeholder communication and management
- Service reporting and governance
- Continuous improvement
- Process audit
- Documentation and reporting
- Adobe Commerce platform updates



The support model works seamlessly with clear accountability distributed across client teams and L1, L2, L3 teams, with a proven governance and communication model.

Service level agreements (SLAS)

The service-level agreement (SLA) defines the level of service expected by a customer from the service provider, with simple metrics by which that service is measured, as well as the remedies or penalties, if any, should the agreed-upon service levels not be achieved.



Issue Type	Impact	Example	Response/Restoration time during supported business hours	
			Gold	Platinum
P1	Critical Impact	Service down or unusable <ul style="list-style-type: none"> Service is unavailable Service is unusable across end user base 	1 hour / 4hours	1 hour / 4 hours
P2	High Impact	Service severely impaired <ul style="list-style-type: none"> Service performance is degraded Service available, but producing significant error messages 	2 hours / 12 hours	2 hours / 8 hours
P3	Medium Impact	Service partially impaired <ul style="list-style-type: none"> Error messages generated, no noticeable end user impact Questions about features used in customer launch 	8 hours / 16 hours	8 hours / 12 hours

The SLAs for different types of incident criticalities can be contracted, maintained, and measured. Further, the response type can have multiple standards (Gold, Platinum, etc.), based on the level of service required by the customer. A typical SLA metric breakdown with multiple service levels looks something like this graph.

Coverage \ Options	Gold	Platinum
Active alert monitoring	APM Only	Active alert monitoring – APM and Manual
On-Desk Coverage	During in-country business hours only	24*7*365
On Call support	N/A	Available
Triage Coverage	During in-country business hours only	24*7*365
SLA Coverage	During in-country business hours only	24*7*365 for P1 issues Business hours support for all other issues
Third party Triage & Communication	During in-country business hours only	24*7*365
Upgrade Support	Covered	Covered

The coverage options for committed SLAs varies with different types of offering. Typically, the scope of Gold and Platinum support services look something like this.

Type	Measurement	Reporting
1 SLA (P1/P2)		
<ul style="list-style-type: none"> Response ————— First response to requester Restoration ————— Time spend in queue* 		Weekly, monthly and quarterly
2 KPI		
<ul style="list-style-type: none"> Response (P3/P4) ————— First response to requester Restoration (P3/P4) ————— Time spend in queue* RCA (P1/P2) ————— RCA complete within agreed time 		Monthly and quarterly
3 Productivity		
<ul style="list-style-type: none"> Incident trend analysis ————— Open, close, backlog (30/60/90 days) MTTR Trends (P1/P2) ————— Average MTTR, minutes impacted Incident reduction ————— Incident reduction trend Problem ageing ————— Open, close, backlog (30/60/90 days) 		Monthly and quarterly

For L1 functions, typical metrics/KPIs collected and reported are represented above.

Key Performance Measurement	Unit of Measure	Reported Metrics
Velocity	Number	No. of story points the team were able to deliver for the sprint
Sprint Commitment Efficiency	Percentage	Total no. of Story Points committed Vs Delivered for a Sprint
Sprint Burn down	Number	Chart (Report, tracks the completion of work throughout the sprint)
Code Quality	Numbers, Percentage	Complexity, LoC, Violations, Code coverage for the sprint
Requirement volatility	Number	# of requirements change/ total # requirements for the sprint
Defect Density	Percentage	$[\text{No. of valid Defects found} / \text{Total No. of Test cases executed}] * 100$ for the sprint
Test Effectiveness	Percentage	$[\text{Valid Defects raised} / (\text{Valid Defects raised} + \text{Rejected defects})] * 100$ for the sprint
Productivity	Number (trend)	Story points delivered per sprint / capacity

Measurement and reporting of L2 services (enhancements and optional services) are similar to development projects. Performance and progress of the L2 teams are measured by metrics like velocity, code quality, test effectiveness, and productivity.

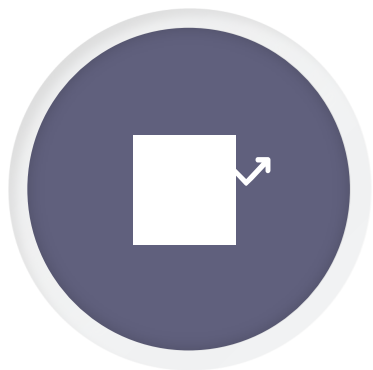
Adobe managed services overview

Adobe Commerce Pro is a platform for delivering ecommerce capabilities, including out-of-the-box features, an ability to customize, and third-party integrations.

Adobe's Managed Services is a hosted and managed application and infrastructure for Adobe Commerce Pro.



Key benefits of Managed Services for Commerce Pro



Decrease Risk With Increased Security

Minimize risk of security threats or downtime with security coverage and a high 99.9% SLA.



Minimize Disruptions & Increase ROI

Simplify and automate tactical day-to-day activities to reduce IT costs and prioritize ROI.



Accelerate Time to Market

Leverage cloud for speed, agility, and control to deliver digital experiences.



Receive Guided Expert Services

Access best practices from a team of cloud experts to deploy and upgrade Commerce Pro

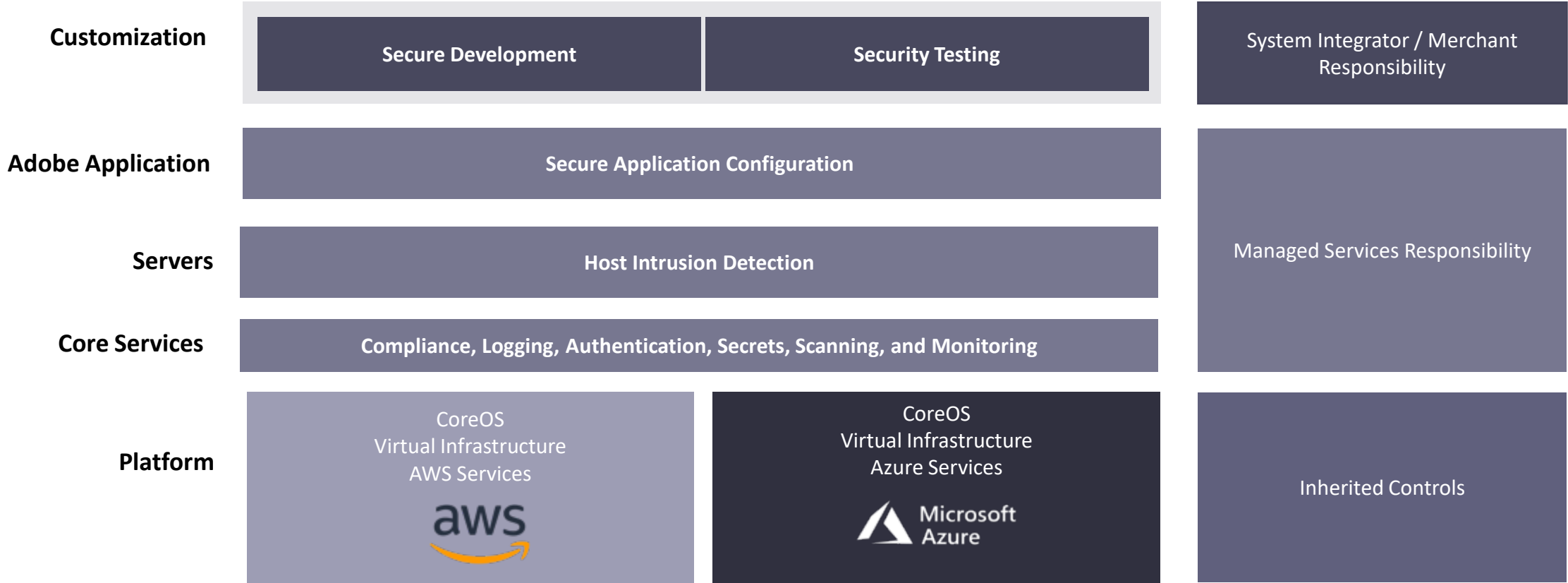


Managed services security technology stack overview

The Adobe Security stack for Managed Services builds security in at every level—using automation and consistency to reduce human error. Development and operations teams automatically inherit security controls from different levels of the stack.

Platform partners such as Amazon Web Services or Microsoft Azure ensures maximum security coverage when applying platform customizations, while Adobe's Managed Services team provides core security services like compliance, logging, authentication, scanning, and monitoring, as well as server security and secure application configuration. Dentsu commerce will be responsible for custom development & integrations, and the associated security processes for custom code.

Managed services security technology stack visualization



Identifying the right support model for your business to continue enhancing and optimizing your platform—and commerce practice as a whole—is a critical step to maintaining all of the hard work that was done during the implementation process. With a comprehensive ongoing support plan in place, your commerce site can keep up with your customers' expectations and you can keep reaching your goals.





About Adobe Commerce

Adobe Commerce, part of Adobe Experience Cloud, is the leading commerce solution for merchants and brands across B2C, B2B and hybrid use cases and was named a leader in the 2020 Gartner Magic Quadrant for Digital Commerce and 2020 Forrester Wave for B2B Commerce. Adobe Commerce, powered by Magento, boasts a strong portfolio of cloud-based omnichannel solutions that empower merchants to successfully integrate digital and physical shopping experiences. Adobe Commerce is the #1 provider to the Digital Commerce 360 Top 1000 online retailers for the tenth consecutive year and the Top 500 Guides for Europe and Latin America. Adobe Commerce is supported by a vast global network of solution and technology partners, a highly active global developer community and a robust eCommerce marketplace for extensions available for download on the Magento Marketplace.

More information can be found at business.adobe.com