

AKAMAI WEB PERFORMANCE FOUNDATIONS



Let's Get to Know Each Other!

Welcome to the [Akamai Web Performance Foundations](#) training!

Please, tell us a few things about yourself:

- What company do you work for?
- What is your role within the company?
- How familiar are you with Akamai products?
- What are your expectations towards this training?



A Few House Rules Before We Start...



Please refrain from checking your mail during the session.



Manage break time optimally.



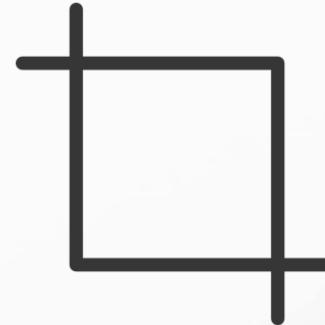
Interact and collaborate with your fellow participants and the trainer.



Ask any questions that you may have at any time.

Screenshots Policy

Some of the screenshots presented in this training have been customized and simplified for your convenience. Whenever in doubt when the content displayed on your screen does not match the one presented in the training, please consult the instructor.



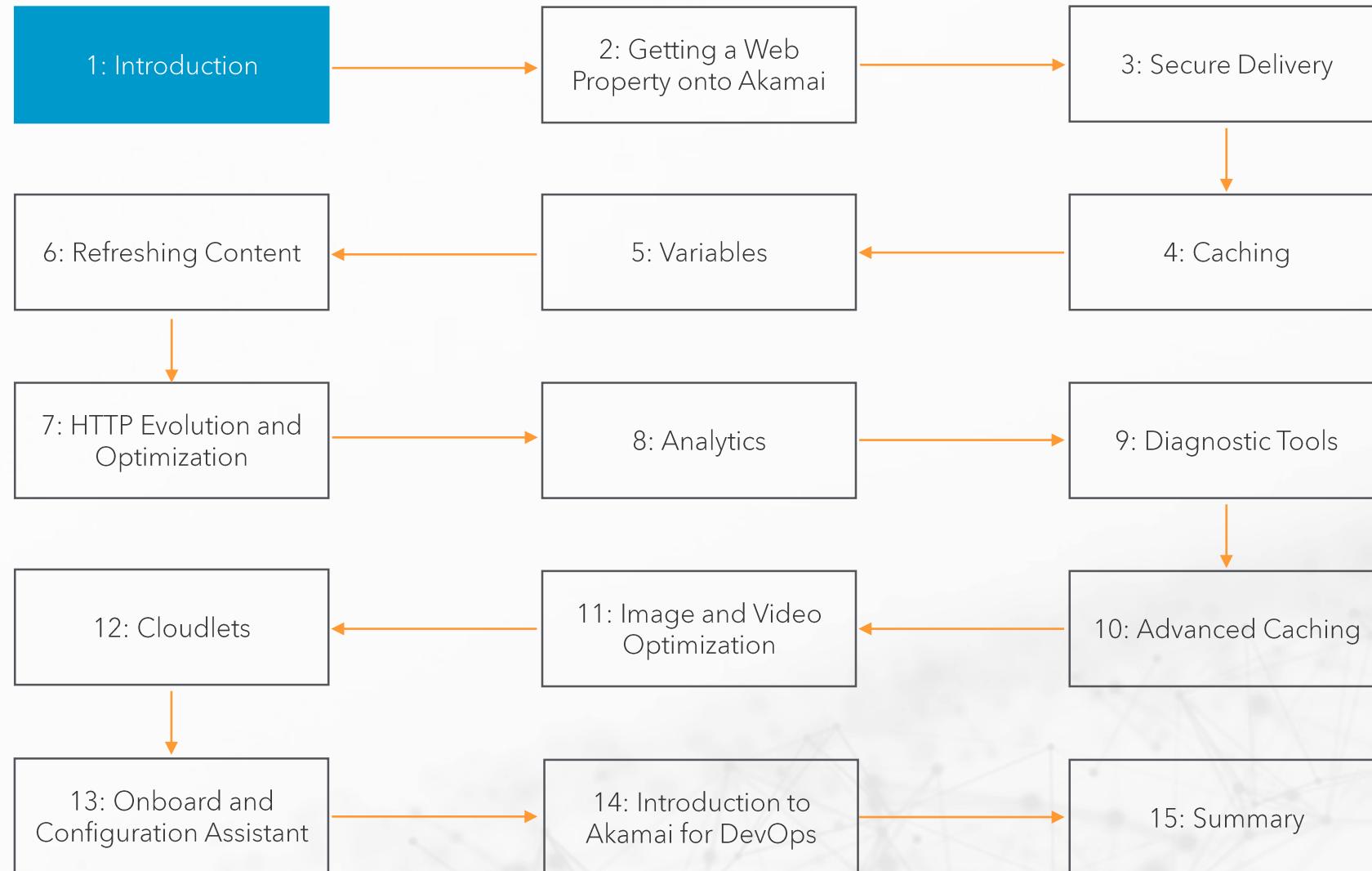
Course Objectives

At the end of this training, you will be able to:

- Bring your site on to the Akamai Intelligent Edge Platform.
- Optimize and tune your site delivery settings.
- Refresh content on Akamai CDN.
- Derive insights into your site using analytics from the Akamai Control Center.
- Automate workflows by leveraging Akamai's APIs and CLIs.



Agenda



MODULE 1

Introduction



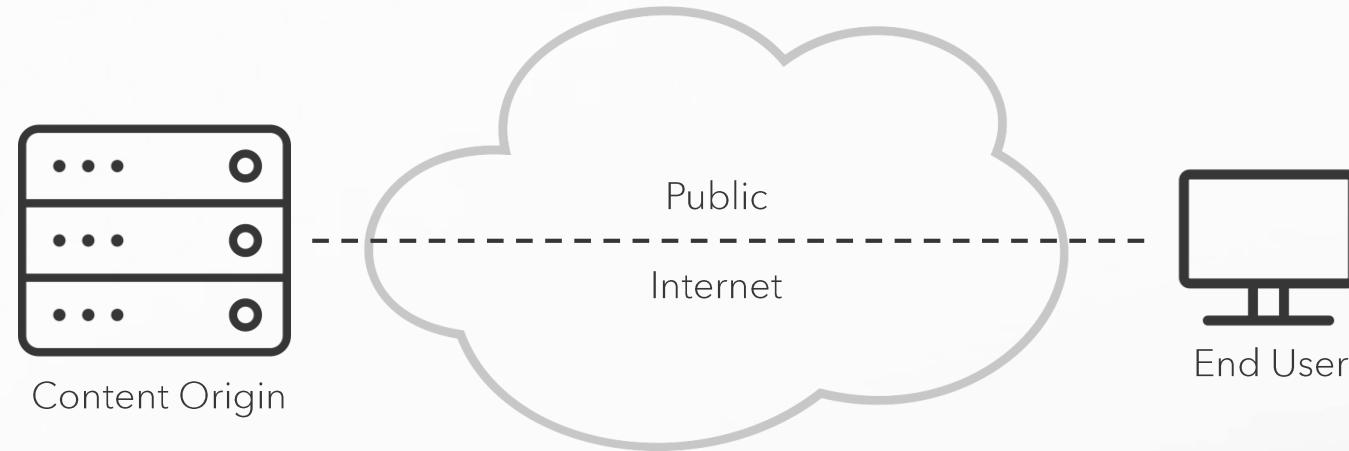
Module Objectives

At the end of this module, you will be able to:

- Describe how the Akamai Intelligent Platform overcomes the challenges of the Internet.
- Describe the Akamai Product Portfolio.
- Describe the products offered in the Akamai Web Performance Solutions portfolio.
- List the features of Akamai Web Performance Solution products.



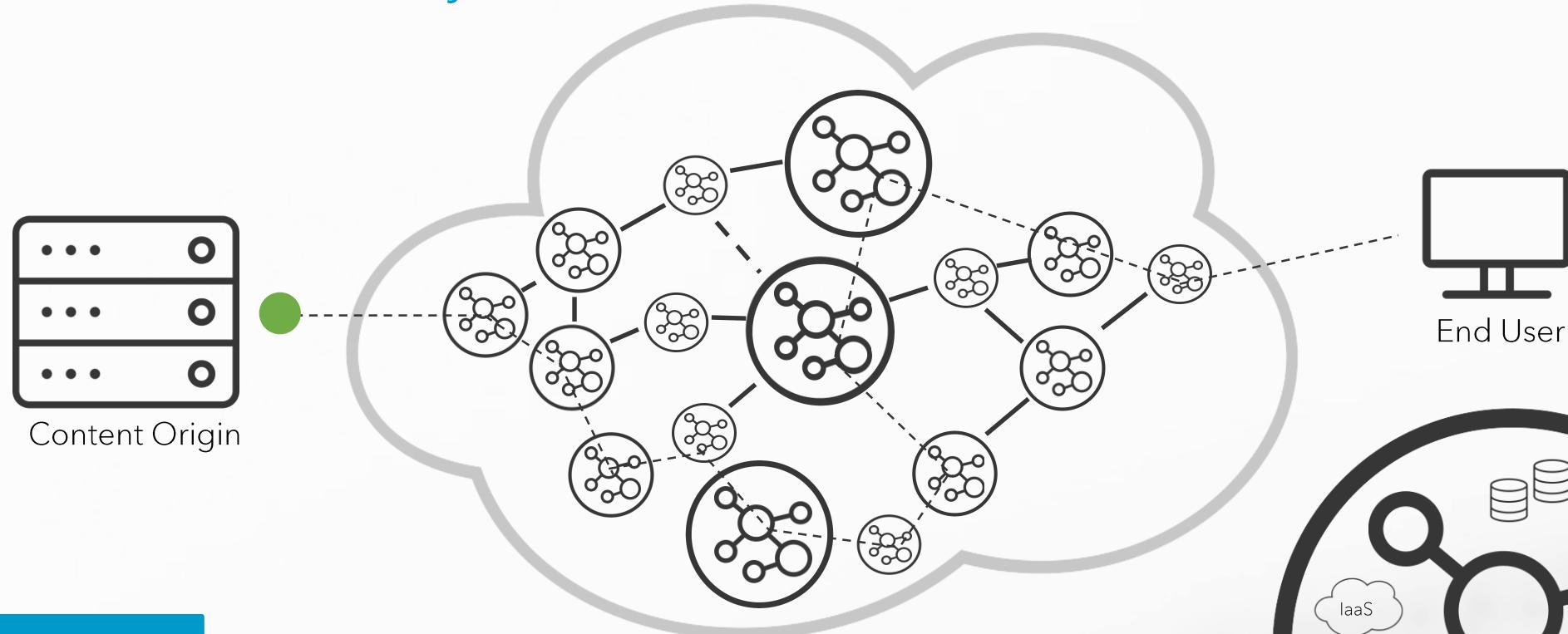
How We Think The Internet Works



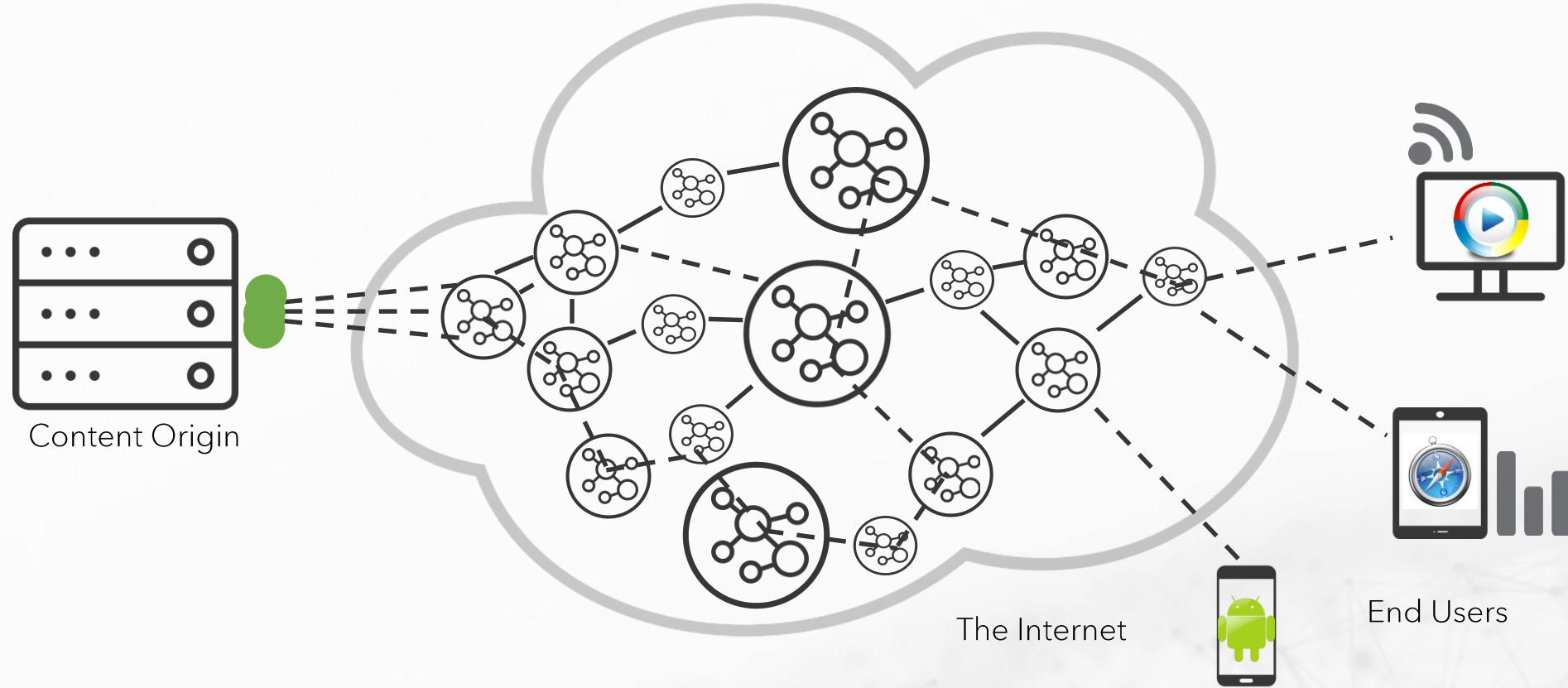
Perception

"It is really simple to get the content in your origin to your end users over the Internet."

How the Internet Really Works

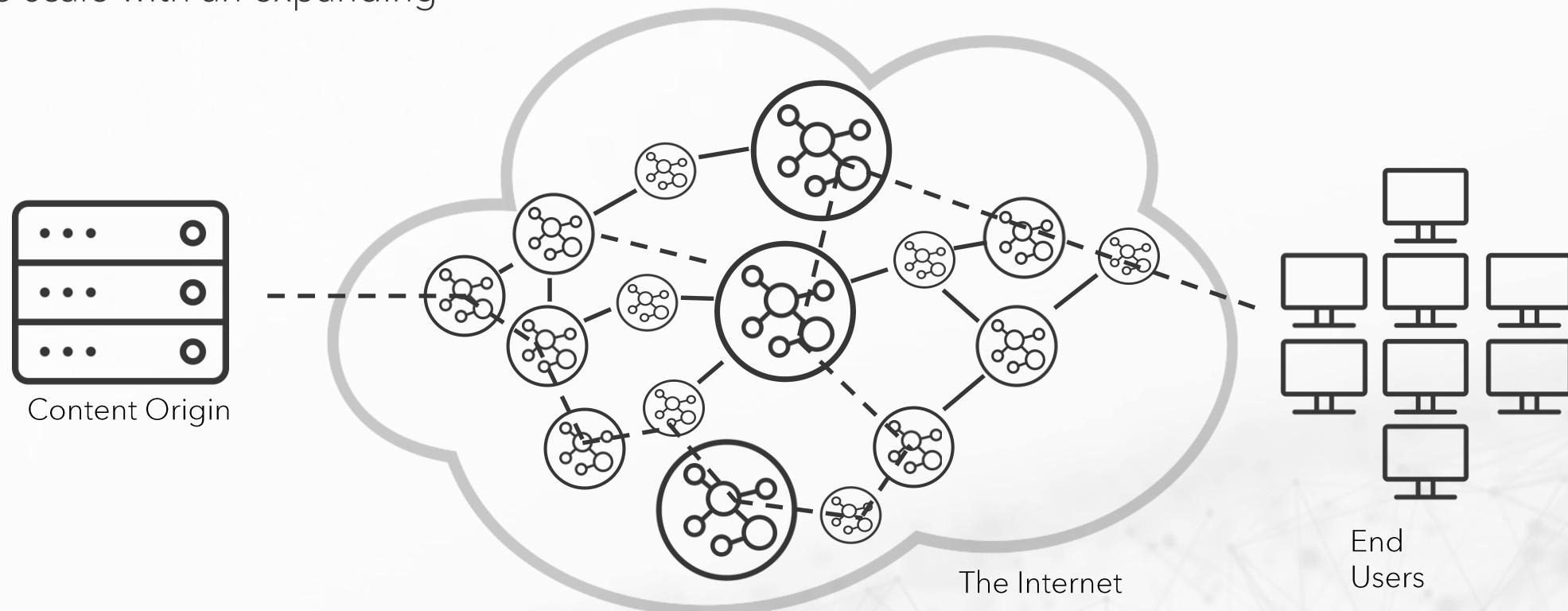


Software, Devices & Connectivity Add Complexity



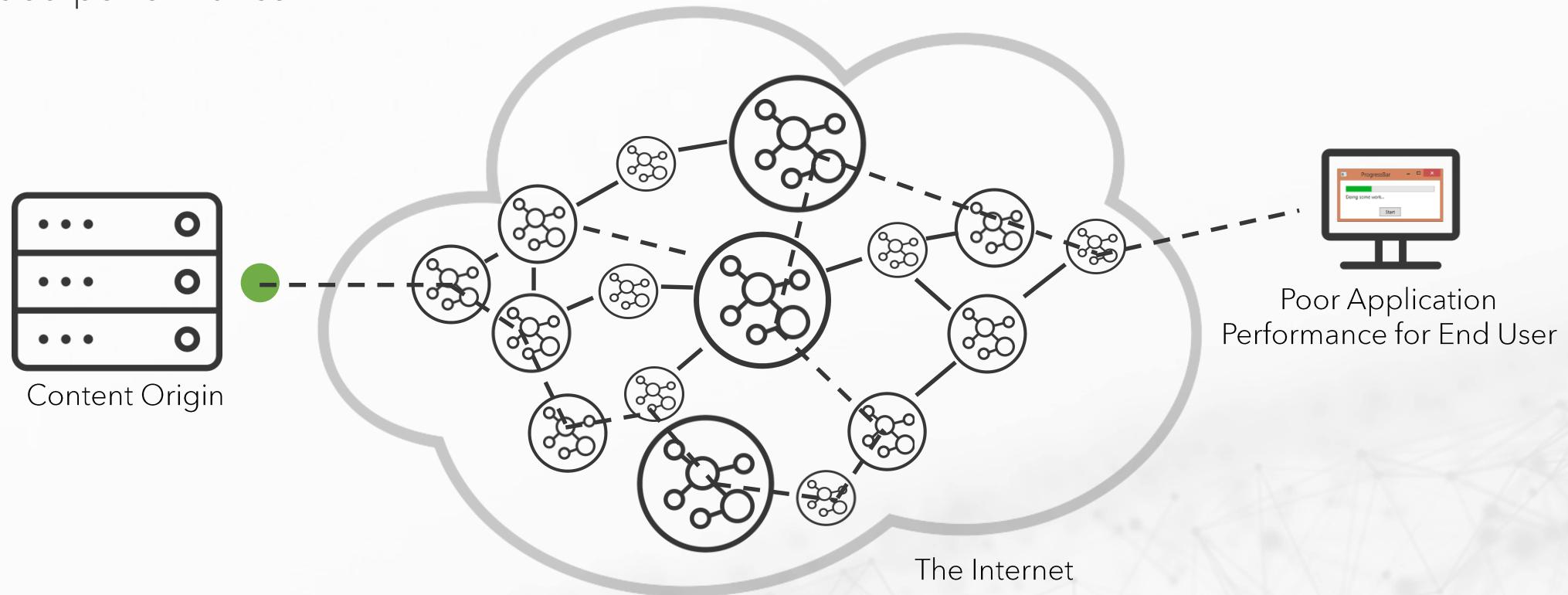
What Do The Complexities Impact?

Inability to scale with an expanding user base



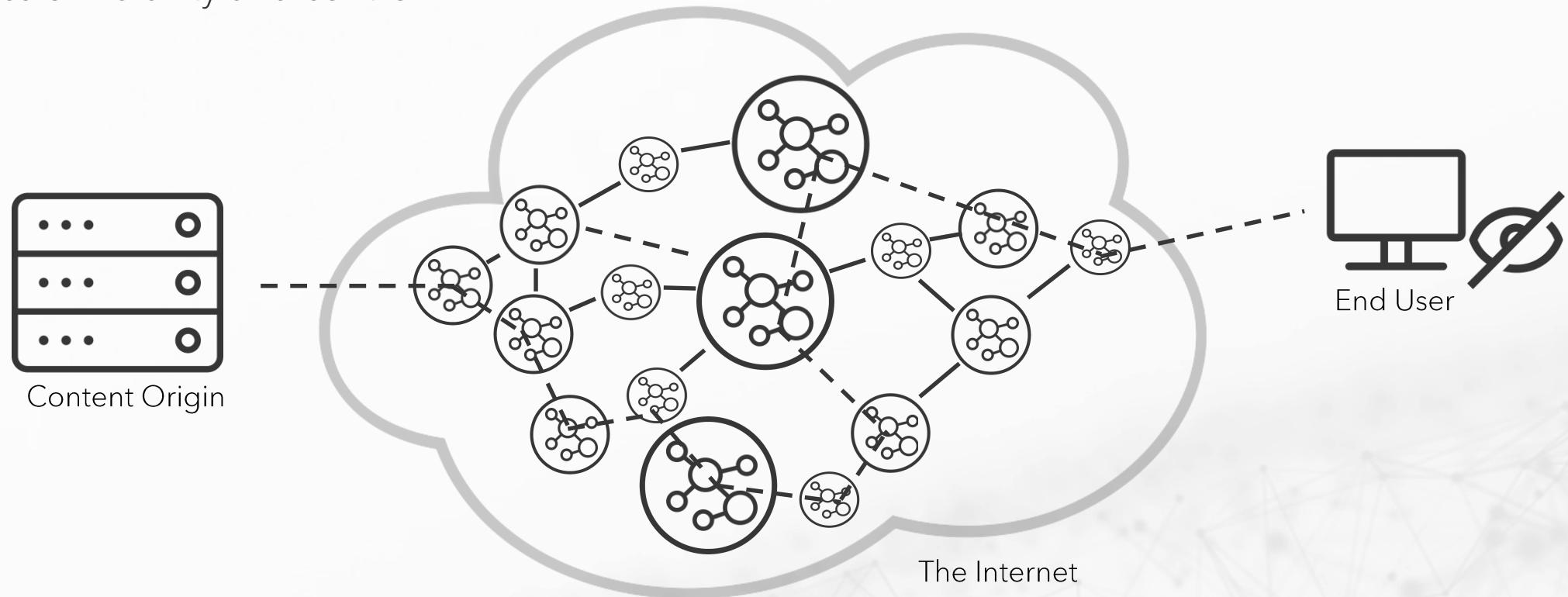
What Do The Complexities Impact?

Downgraded performance



What Do The Complexities Impact?

Overall loss of visibility and control



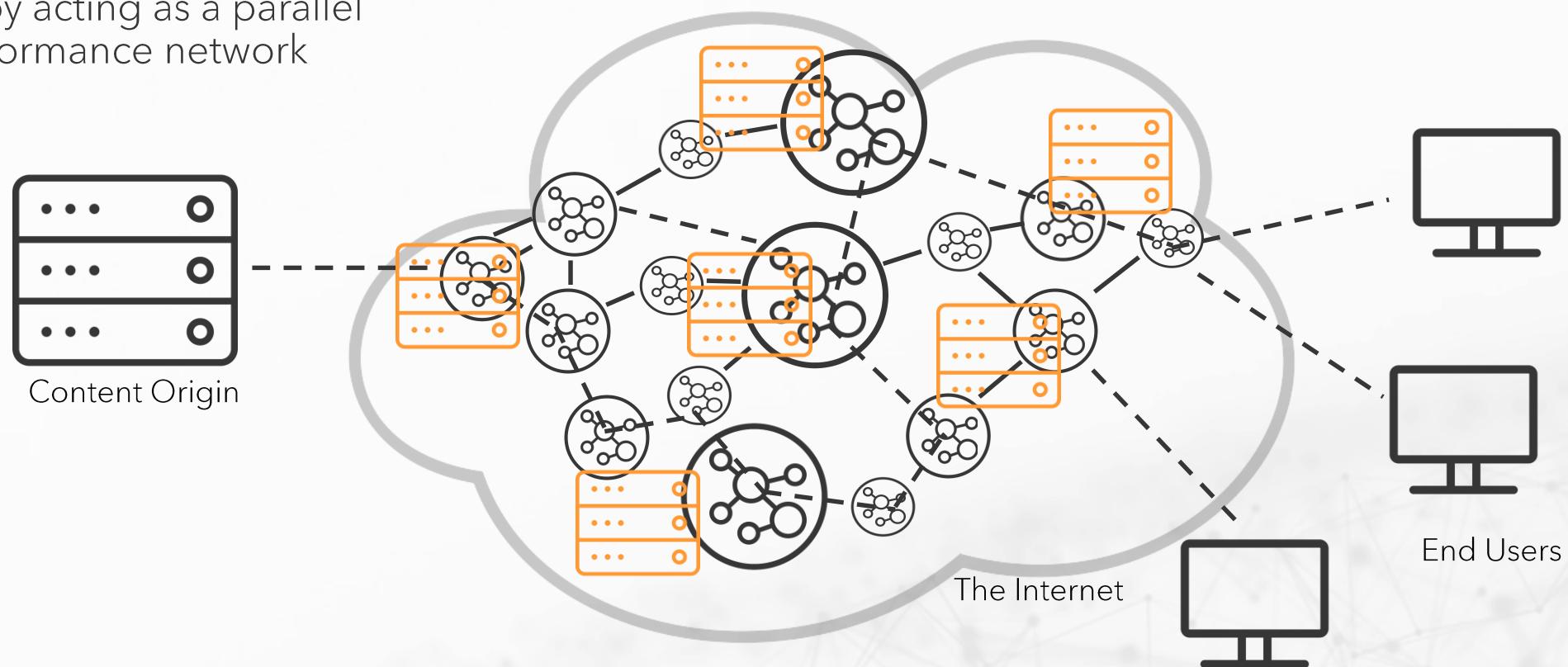
Test Your Akamai Edge!

Select the three factors increasing the complexity of content delivery over the Internet.

- A. Myriad of device types, operating systems, and browsers
- B. Multiple ISP networks, peering points, and data centers
- C. Variety of end user experiences – applications, websites, videos, and so on
- D. Increasing investments in scaling enterprise network infrastructure

The Akamai Intelligent Edge Platform Overlays the Internet

Solves complexities of the Internet by acting as a parallel high-performance network

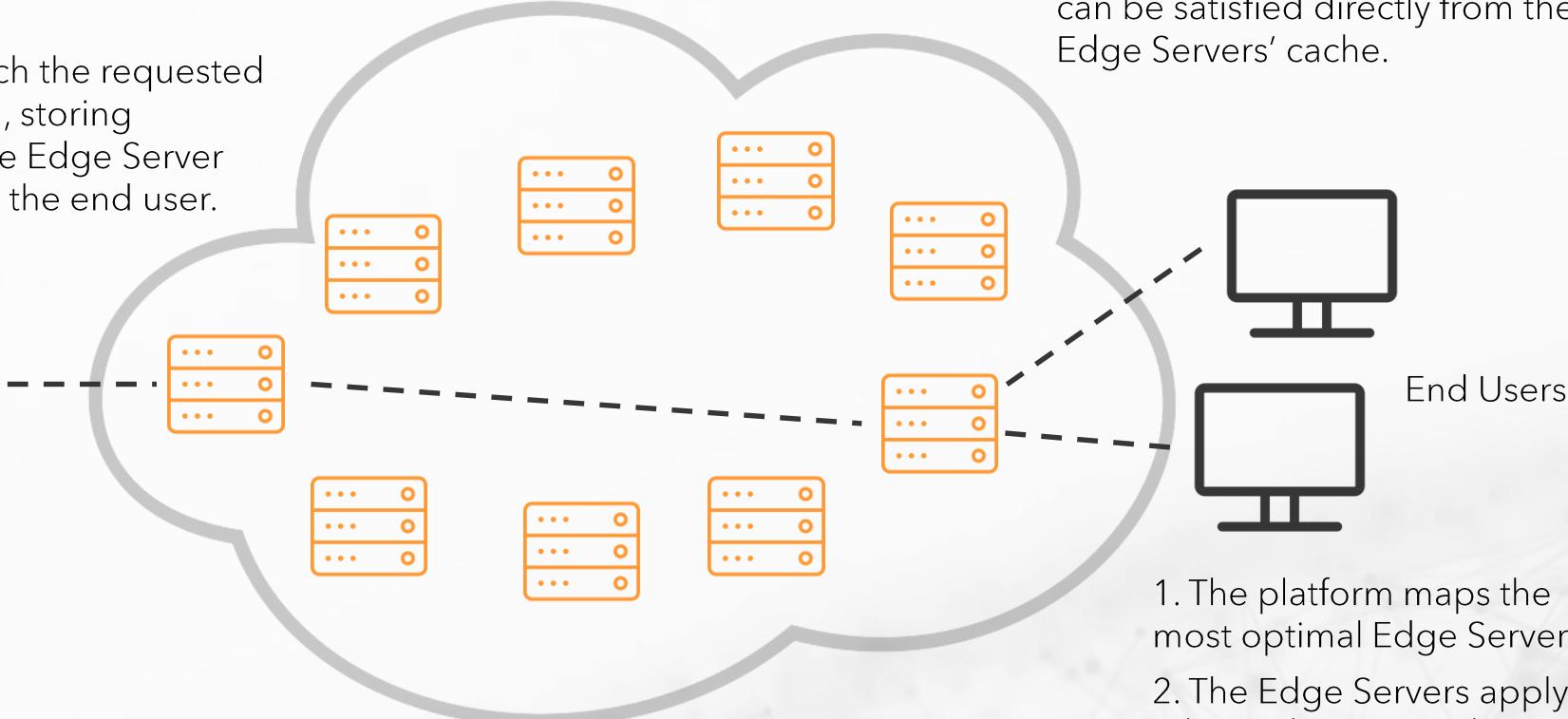


How the Akamai Intelligent Edge Platform Works for Cached Content

3. The Edge Servers fetch the requested content from the Origin, storing cacheable content in the Edge Server caches, and deliver it to the end user.



Content Origin



4. Future requests for the same content can be satisfied directly from the Akamai Edge Servers' cache.

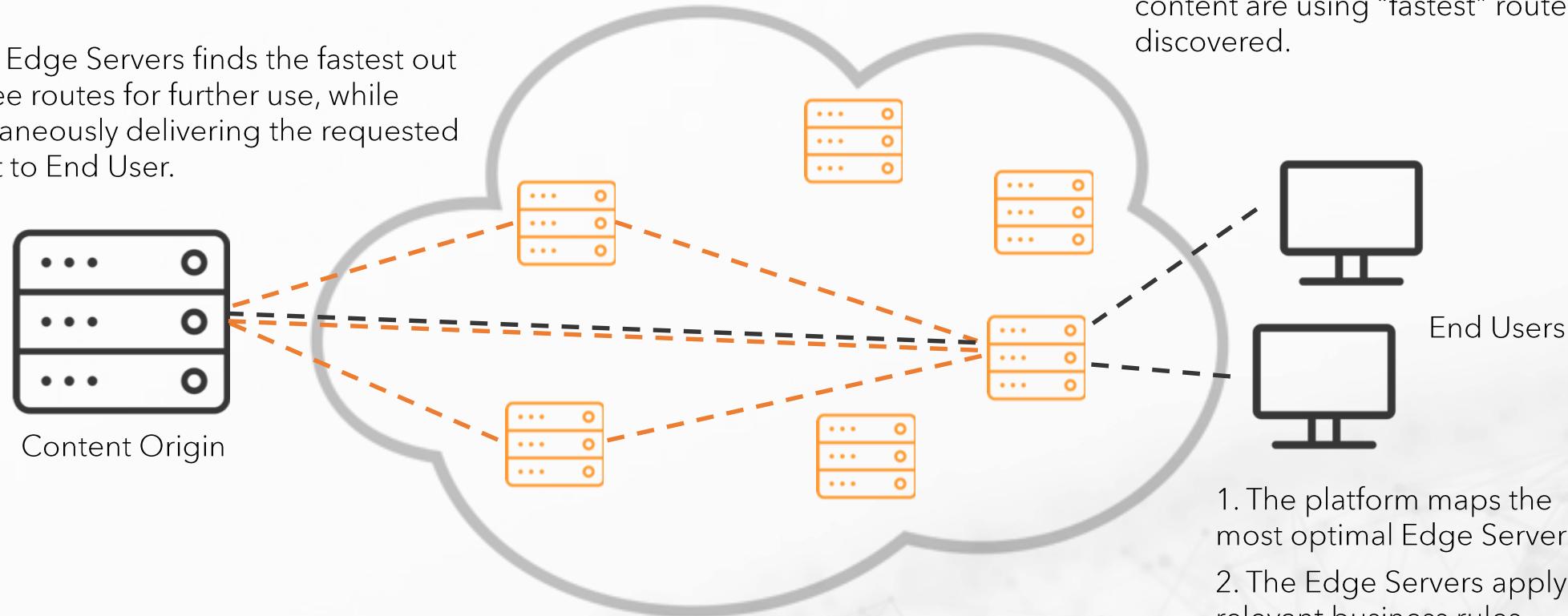


End Users

1. The platform maps the most optimal Edge Server.
2. The Edge Servers apply the relevant business rules configured by the customer.

How the Akamai Intelligent Edge Platform Works for Non-Cached Content

3. The Edge Servers finds the fastest out of three routes for further use, while simultaneously delivering the requested object to End User.



Test Your Akamai Edge!

Select the sequence that illustrates how the Akamai Intelligent Edge Platform works for cached content.

1. Subsequent requests for the same content are served from the Edge server cache.
 2. The platform maps the most optimal Edge server to the end user.
 3. The Edge server applies business rules set by the customer.
 4. The Edge server fetches content and authorization from the origin and delivers to the end user.
- A. 1-2-3-4
- B. 2-3-4-1
- C. 3-1-4-2
- D. 4-2-1-3

Akamai Solutions Portfolio

AKAMAI SOLUTIONS

WEB PERFORMANCE SOLUTIONS

Accelerate websites to grow revenue and conduct business globally—on any device, anywhere

MEDIA DELIVERY SOLUTIONS

Simplify workflow and deliver the highest quality media affordably and at scale

CLOUD SECURITY SOLUTIONS

Secure websites and data centers to reduce the risk of downtime and data theft

CLOUD NETWORKING SOLUTIONS

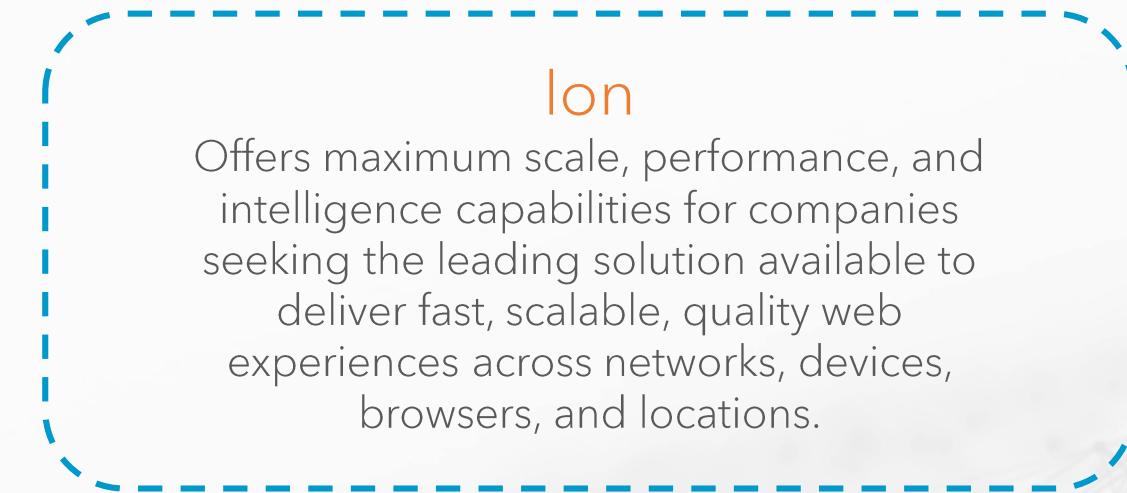
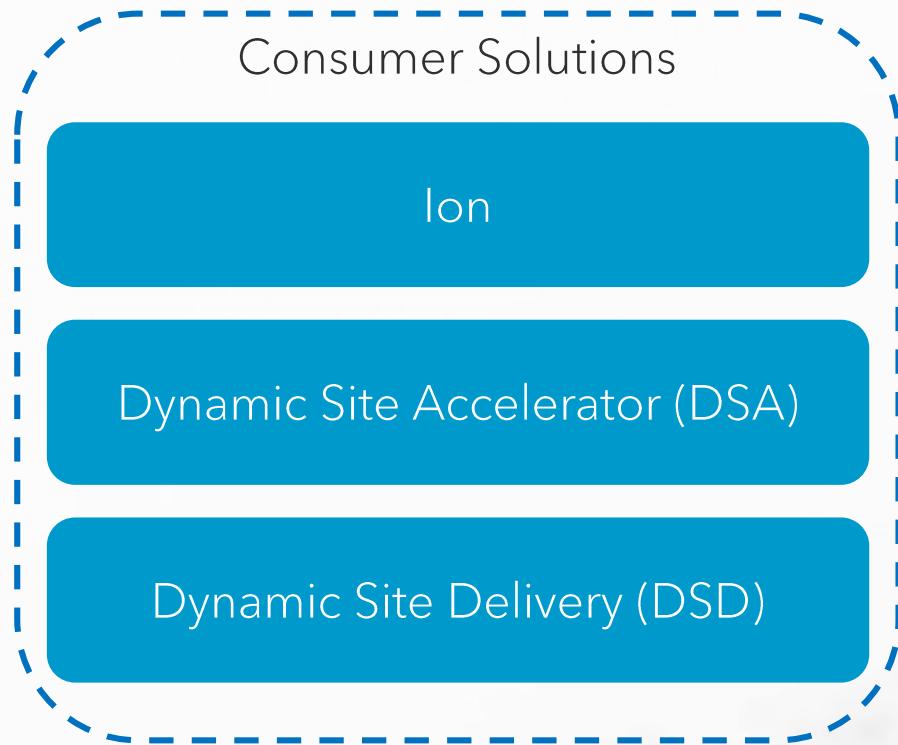
Transform the enterprise network to accelerate applications, lower costs, and connect to clouds

NETWORK OPERATOR SOLUTIONS

Optimize network traffic, enable new revenue streams and control costs

SERVICES & SUPPORT

Akamai Web Performance Solutions Portfolio - Core Site-Delivery Products



Akamai Web Performance Solutions Portfolio - Additional Web Performance Products

- Image Manager
- Cloudlets
- Cloud Monitor
- Global Traffic Manager
- API Gateway
- FastDNS (technically a Security product)
- mPulse
- CloudTest
- DevOps

Products vs Features

Products are things you purchase and features are sub-components of those products.

Example: Ion contains many features, including:

- SureRoute
- Object Prefetching
- Adaptive Acceleration
- Device Characterization
- Dynamic Page Caching
- ...and much more

Q&A



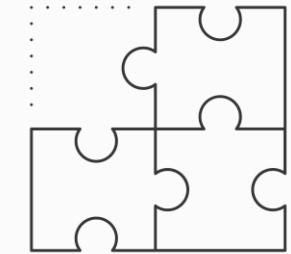
Summary

Complexities posed by the Internet include:

- Inability to scale
- Degraded performance
- Overall loss of visibility & control

The Akamai Intelligent Edge Platform provides:

- Availability
- Scalability
- Reliability
- 240+ edge servers globally distributed



The Akamai Solutions Portfolio comprises:

- Web Performance Solutions
- Media Delivery Solutions
- Cloud Security Solutions
- Cloud Networking Solutions
- Network Operator Solutions
- Services and Support

Core Web Delivery Products

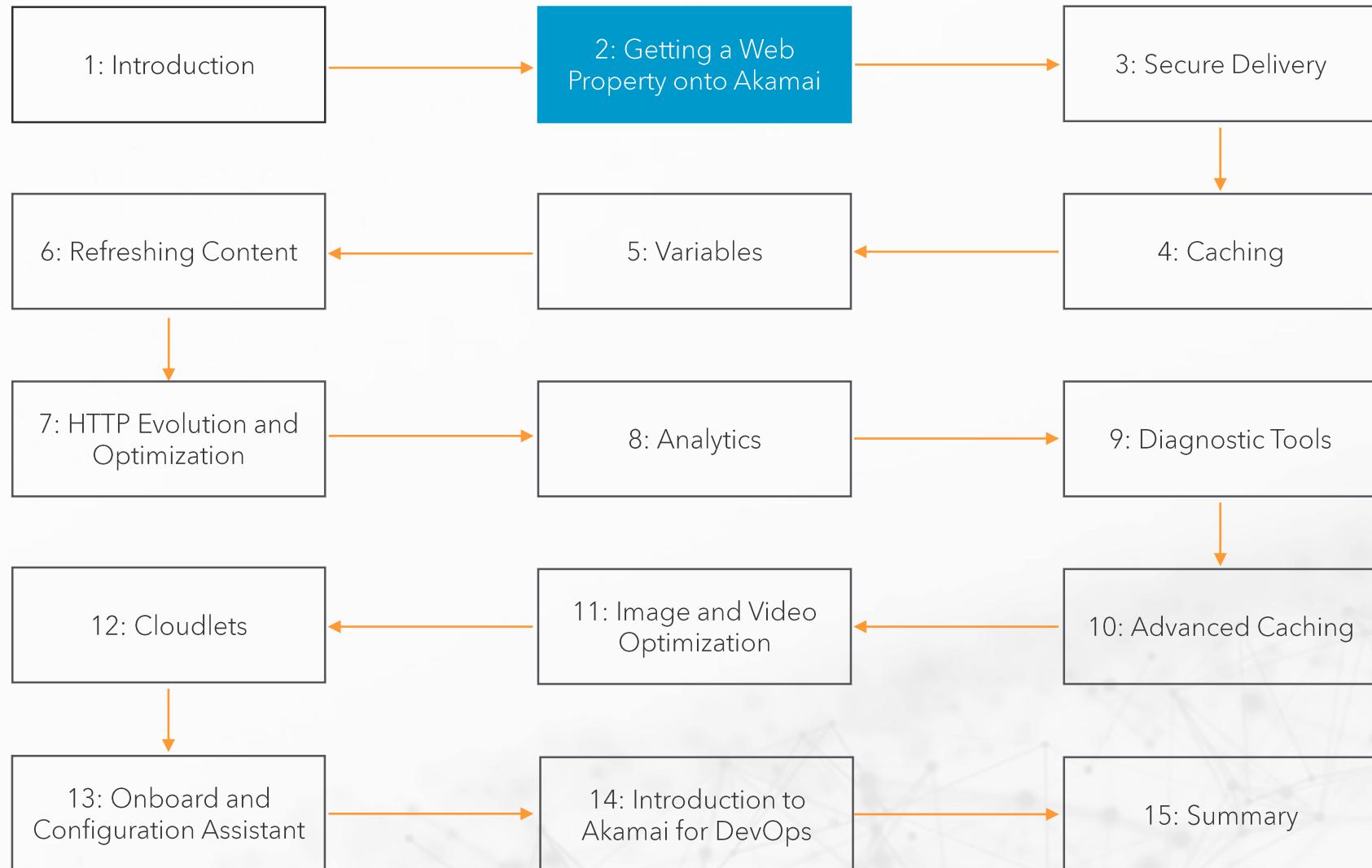
Akamai's Core Site-Delivery Products Include:

- Ion
- Dynamic Site Accelerator
- Dynamic Site Delivery

Complexities posed by the Internet include:

- Inability to scale
- Degraded performance
- Overall loss of visibility & control

Agenda



MODULE 2

Getting a Web Property onto
Akamai



Module Objectives

At the end of this module, you will understand:

- DNS record and hostname fundamentals.
- How DNS is used to get a web site onto Akamai.
- The Akamai Staging network.
- The Akamai Control Center control center and using Property Manager.
- CP Codes.



DNS Record Types

A, AAAA Record

Points a DNS hostname directly to an IP address

www.example.com.	11984	IN	A	93.184.216.34
www.example.com.	86176	IN	AAAA	2606:2800:220:1:248:1893:25c8:1946

CNAME Record

Points a DNS hostname directly to another hostname

www.akamai.com.	290	IN	CNAME	www.akamai.com.edgekey.net.
-----------------	-----	----	-------	--

DNS Record Types

A Record

Points a DNS hostname directly to an IPv4 address

```
dig A www.example.com
```

AAAA Record

Points a DNS hostname directly to an IPv6 address

```
dig AAAA www.example.com
```

```
dig A www.example.com

; <>> DiG 9.10.6 <>> A www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55593
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
;; ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 8192
;; QUESTION SECTION:
;www.example.com. IN A

;; ANSWER SECTION:
www.example.com. 7524 IN A 93.184.216.34
```

DNS Record Types

CNAME Record

Points a DNS record directly to another hostname.

```
dig CNAME www.akamai.com
```

```
dig CNAME www.akamai.com
```

```
; <>> DiG 9.10.6 <>> CNAME www.akamai.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20556
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
;; ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 8192
;; CLIENT-SUBNET: 88.221.209.14/32/0
;; QUESTION SECTION:
;www.akamai.com. IN CNAME

;; ANSWER SECTION:
www.akamai.com. 300 IN CNAME www.akamai.com.edgekey.net.
```

Hostnames

Hostname (ex www.akamai.com) serves two functions:

```
[krk-mp64s:~ marcinj$ curl https://www.akamai.com/pl/pl -vs -o /dev/null -4
* Trying 92.123.14.253...
* TCP_NODELAY set
* Connected to www.akamai.com (92.123.14.253) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
* Server certificate: www.akamai.com
* Server certificate: DigiCert ECC Secure Server CA
* Server certificate: DigiCert Global Root CA
> GET /pl/pl HTTP/1.1
> Host: www.akamai.com
> User-Agent: curl/7.60.0-DEV
> Accept: */*
>
< HTTP/1.1 200 OK
< Content-Length: 65338
< Content-Type: text/html; charset=UTF-8
< Set-Cookie: JSESSIONID=49DE96A609D474828B31C734EA15ADC7; path=/; Secure; HttpOnly
```

An address:

The hostname points in DNS to an address, which tells the client where to go for the content.

A Host Header:

As the value of HTTP Host header tells the server, which site to serve.

Akamai Edge Hostnames - Activity

Take a moment to resolve the DNS for one of your company's sites / applications.

Ways you can do this:

- You can use nslookup, dig, or similar tools on your laptop
- Use the DNS lookup tool available within Akamai Control Center by going to RESOLVE->Diagnostic Tools and then choosing Get Domain Details (DIG)
- Google "online dig" for an online DNS resolution tool

Things to think about:

- Is your site on edgesuite or edgekey?
- Is there anything else interesting going on in the DNS resolution chain?

Akamai Edge Hostnames

Akamai Edge Hostnames are special Akamai hostnames that you CNAME your property to in order to get it onto the Akamai network.



Once the DNS for a site points to an Akamai Edge Hostname, Akamai's mapping algorithms determine the exact IP address of an optimal Edge Server to resolve to.

Akamai Edge Hostnames

Akamai Edge Hostnames are usually from one of two networks:

[customer-hostname].edgesuite.net (Akamai Edgesuite aka Free-Flow Network)

[customer-hostname].edgekey.net (Akamai Edgekey aka Secure Network)

Do both of these networks provide HTTP and HTTPS? What do you think?

How a Site is Akamized

Let's take an example of an Akamaized site:

```
dig www.akamai.com
;; ANSWER SECTION:
www.akamai.com.          294      IN      CNAME    www.akamai.com.edgekey.net.
www.akamai.com.edgekey.net. 21594    IN      CNAME    www.akamai.com.edgekey.net.globalredir.akadns.net.
www.akamai.com.edgekey.net.globalredir.akadns.net. 3594    IN      CNAME    e1699.dscx.akamaiedge.net.
e1699.dscx.akamaiedge.net. 14       IN      A       92.123.14.253
```



Akamai Staging Network

Akamai Staging Network is a secondary network for testing purposes.

- When activating a version of property (configuration) on the Akamai Network, you can activate it either on Staging or on Production.
- It is recommended that you always activate on Staging first, test your site on the Staging network, and then activate on Production once satisfied.
- Staging network is not meant to be used for performance testing. This is for functional testing only.

Staging Edge Hostnames

Akamai Staging Network equivalent Edge Hostnames are:

[customer-hostname].edgesuite-staging.net (Akamai Edgesuite Network)

[customer-hostname].edgekey-staging.net (Akamai Edgekey Network)

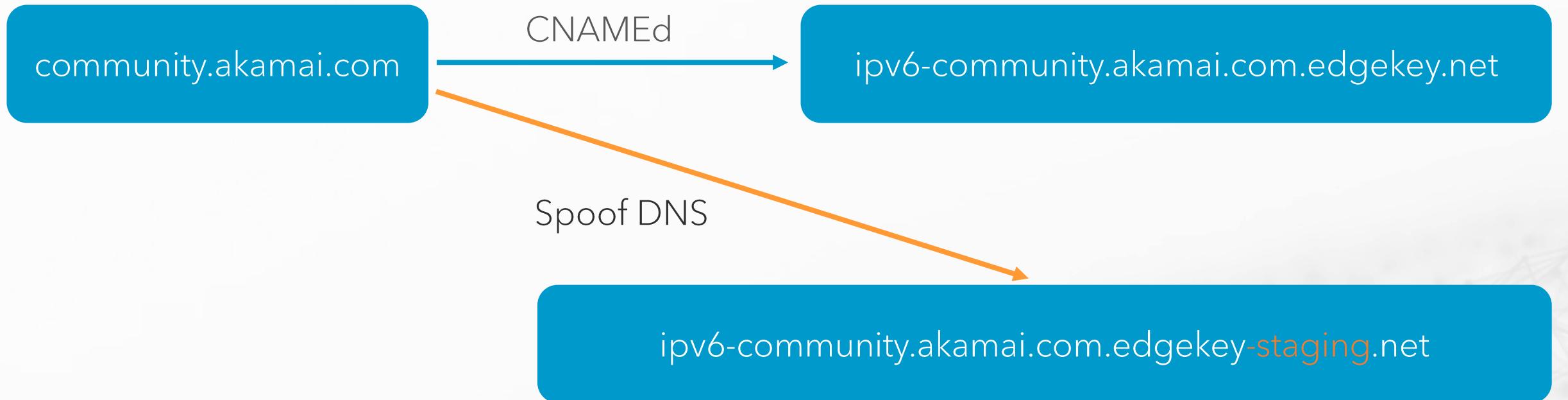
Test Your Akamai Edge!

What is the main purpose of the Akamai Staging network?

- A. For Akamai to test out their latest product/feature updates
- B. For you to sync with your own staging/dev/qa environments
- C. For you to test out configuration changes to your Akamai properties prior to going live
- D. For prospective Akamai customers to try out Akamai before they decide to buy into the platform

Testing on the Staging Network

Spoof your DNS to the Staging version of your Edge Hostname.



Testing on the Staging Network

2. Add *-staging* to your Edge Hostname

ipv6-community.akamai.com.edgekey.net

Becomes



ipv6-community.akamai.com.edgekey-
staging.net

Testing on the Staging Network

To test you need to spoof your DNS to the Staging version.
Your DNS should resolve your hostname (ex www.akamai.com) to staging IP address.

Example

```
community.akamai.com. 25 IN CNAME ipv6-community.akamai.com.edgekey.net.
ipv6-community.akamai.com.edgekey.net. 16954 IN CNAME e3953.dsrb.akamaiedge.net.
e3953.dsrb.akamaiedge.net. 20 IN CNAME e3953.dsrb.akamaiedge.net.0.1.cn.akamaiedge.net.
e3953.dsrb.akamaiedge.net.0.1.cn.akamaiedge.net. 20 IN A 104.103.76.115
```

Staging IP should be retrieved from resolving **ipv6-community.akamai.com.edgekey-staging.net.**

```
ipv6-community.akamai.com.edgekey-staging.net. 21600 IN CNAME e3953.dsrb.akamaiedge-staging.net.
e3953.dsrb.akamaiedge-staging.net. 20 IN A 23.44.114.119
```

We end up with record:

```
23.44.114.119 community.akamai.com
```

Testing on the Staging Network

Spoofing can be achieved by adding an entry to your hostfile, or using an HTTP proxy such as Charles or Fiddler.

Windows

On Windows, the hostfile is:

C:\Windows\System32\drivers\etc\hosts

Mac

On Mac, the hostfile is: /etc/hosts

Remember:

Corporate proxies can make resolution based on Host header, so they will prevent your efforts to spoof the address.

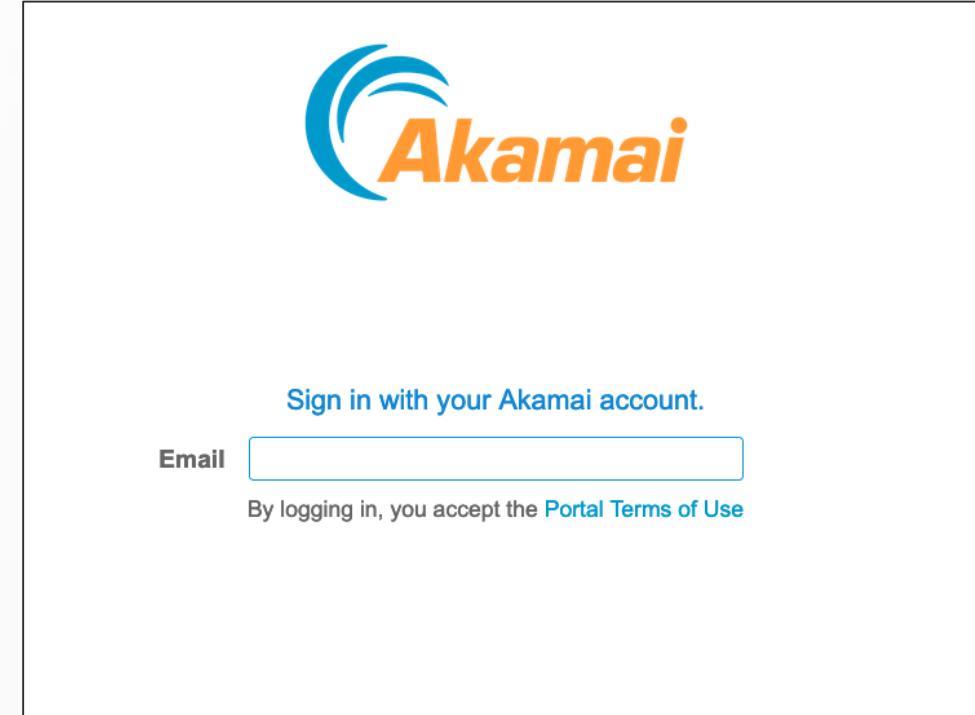
Question:

How to verify that you are testing against the staging network?

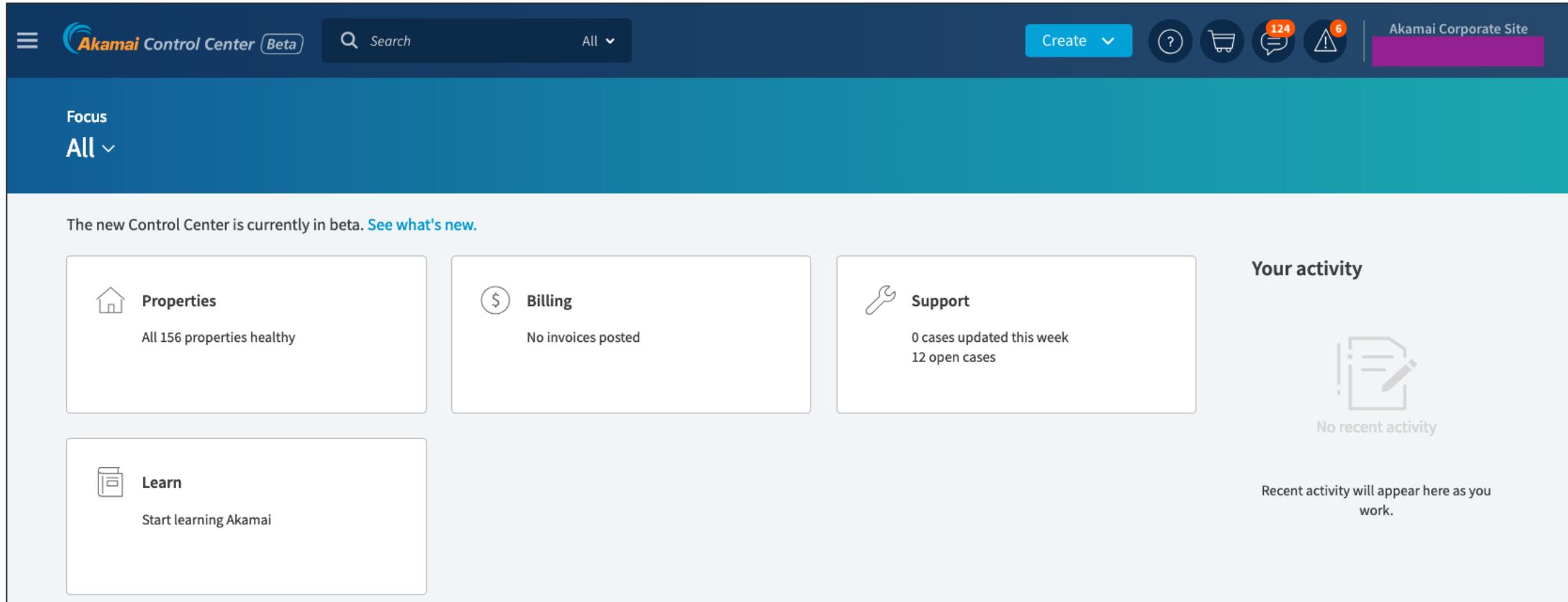
Akamai Control Center

Akamai Control Center is the main portal for configuration, reporting, purging content, etc.

<https://control.akamai.com/>



Navigating Akamai Control Center



The new Control Center is currently in beta. [See what's new.](#)

**Properties**
All 156 properties healthy

**Billing**
No invoices posted

**Support**
0 cases updated this week
12 open cases

Your activity

No recent activity

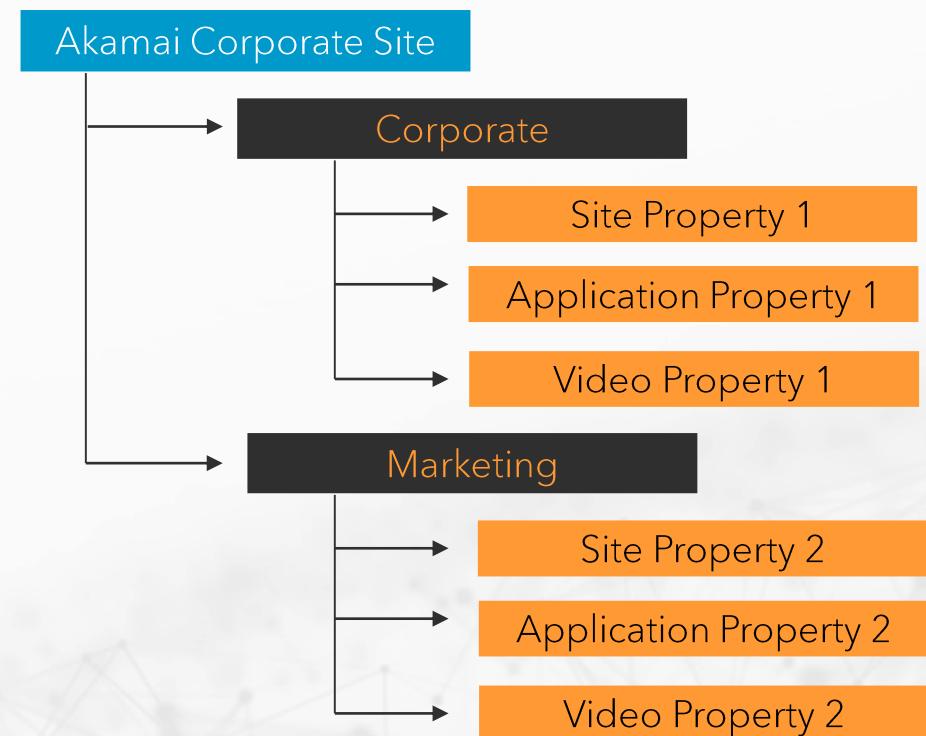
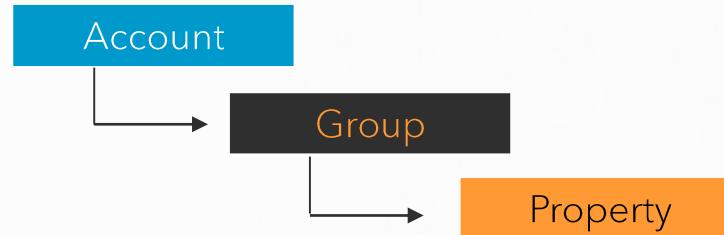
Recent activity will appear here as you work.

**Learn**
Start learning Akamai

Akamai Control Center Content Model

Three major components:

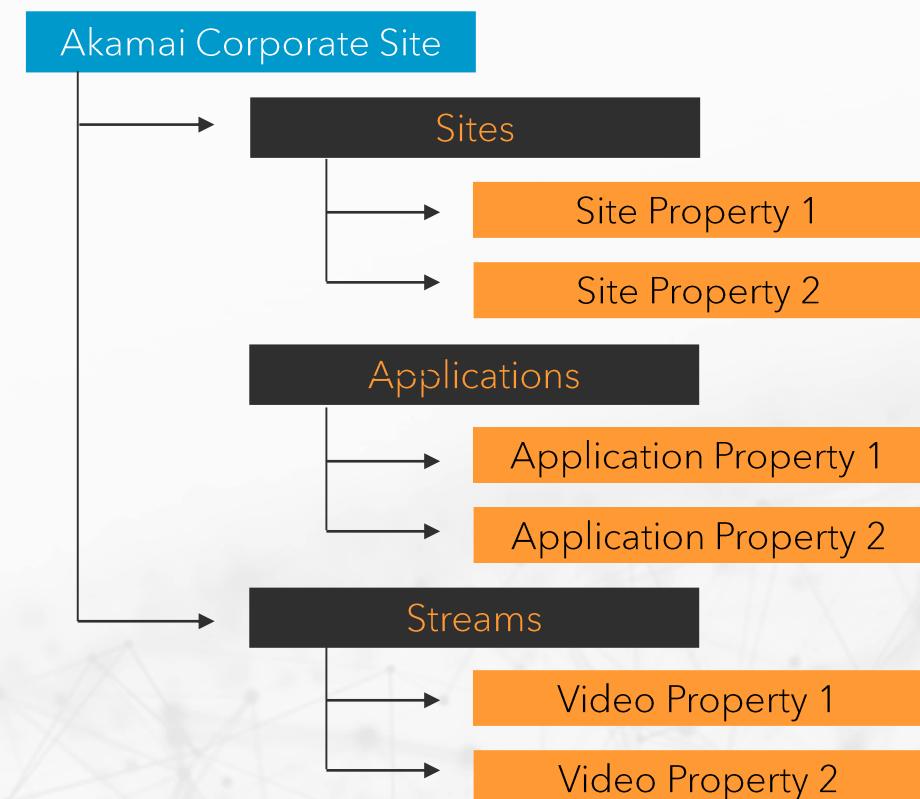
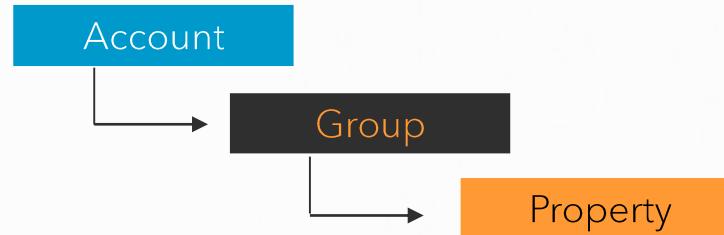
Content organized by Business Organization



Akamai Control Center Content Model

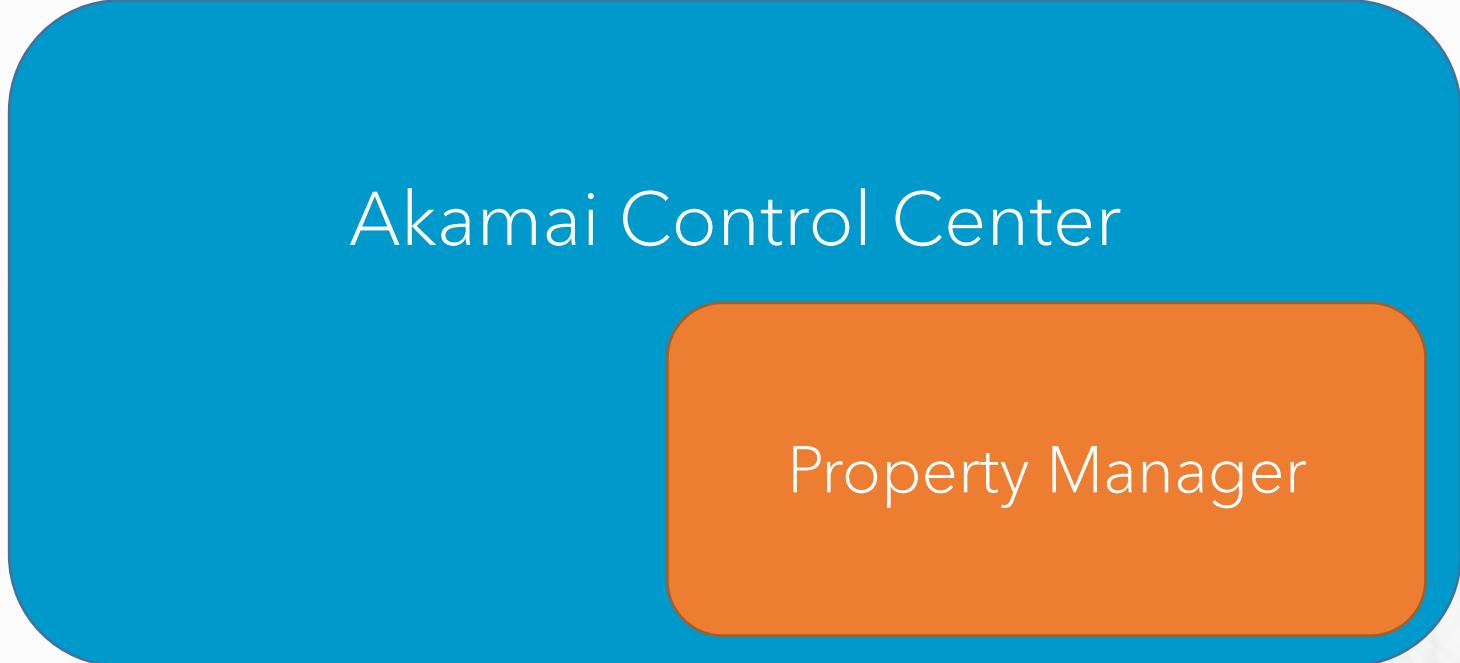
Three major components:

Content organized by Technologies



Property Manager

Property Manager is tool for creating and managing web properties within the Akamai Control Center.

A diagram illustrating the Akamai Control Center. It features a large blue rounded rectangle labeled "Akamai Control Center". Inside this, a smaller orange rounded rectangle is labeled "Property Manager", indicating its central role within the system.

Akamai Control Center

Property Manager

Property Manager

Akamai Control Center Search All Create ? 123 6 Akamai Corporate Site

Identity and Access Management

Help

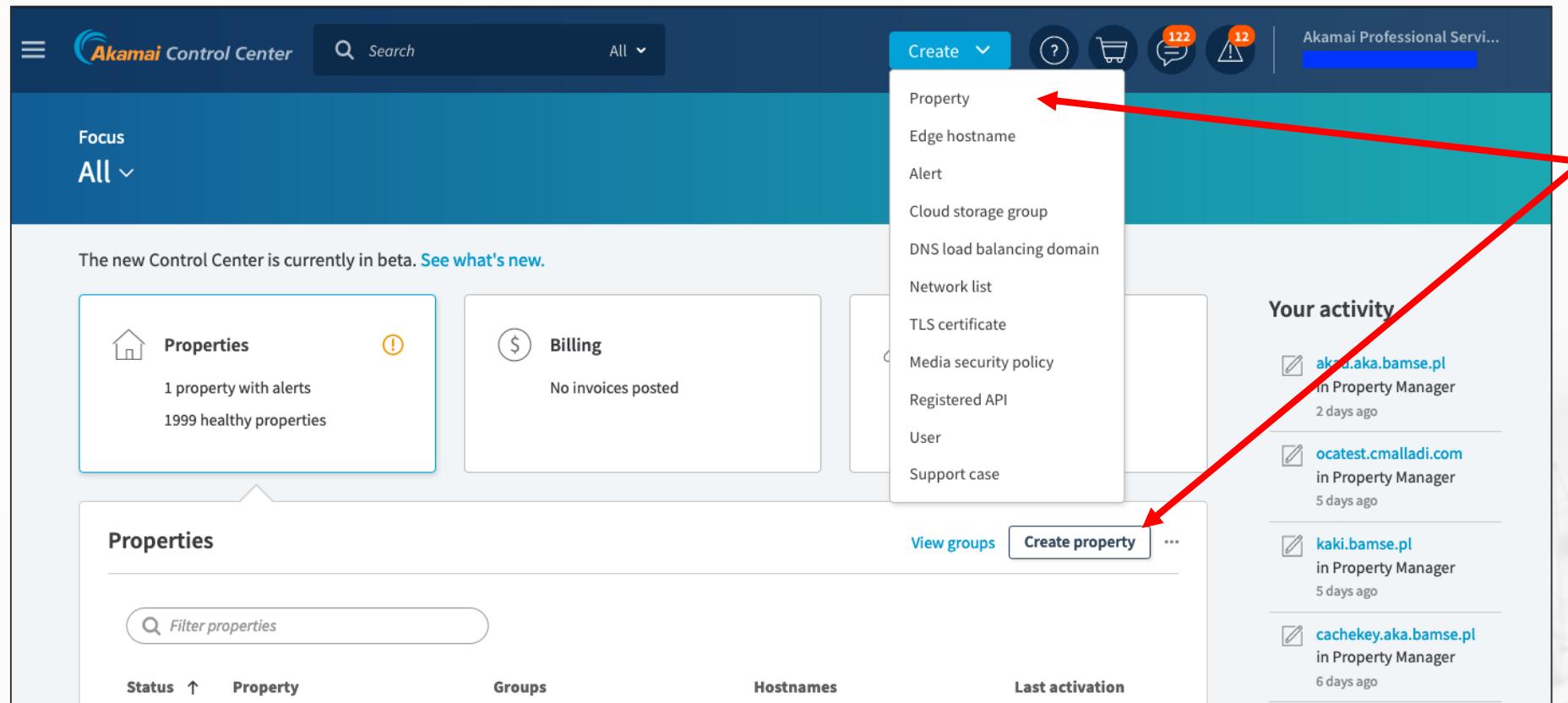
Users and API Clients Roles **Groups** Escalation Contacts Properties Resources Settings

Organize your groups and subgroups. This tab lets you create groups and move them within the group hierarchy structure. Access to groups and resources is based on the nested structure of the group hierarchy, and the roles each user has for a group. Moving a group within another group may affect users' access to the resources in the moved group.

Group name	Action
<input type="text"/> Filter	
▼ Akamai Corporate Site-1-3CROXN	...
dev-wwwnew	...
Acquisitions	...
All Hands	...
Live Webinar - Jul 2014	...
Marketing Events	...

Property Manager

Creating a new Web Property with Property Manager



The new Control Center is currently in beta. [See what's new.](#)

Properties 1 property with alerts 1999 healthy properties

Billing No invoices posted

Properties

Groups **Hostnames** **Last activation**

Create ▾

- Property →
- Edge hostname
- Alert
- Cloud storage group
- DNS load balancing domain
- Network list
- TLS certificate
- Media security policy
- Registered API
- User
- Support case

Create property ...

Your activity

- akad.aka.bamse.pl in Property Manager 2 days ago
- ocatest.cmalladi.com in Property Manager 5 days ago
- kaki.bamse.pl in Property Manager 5 days ago
- cachekey.aka.bamse.pl in Property Manager 6 days ago

Create Property

Property Names

- The Property Name is not the site domain / hostname, it is just a label.
- It is a good idea to give your property an appropriate name.

New Property

Create a new property to control how Akamai edge servers manage traffic to your site.

Product Type: **Property Manager**

Product: Cloud Monitor Data Delivery
 Dynamic Site Delivery
 Ion Premier
 Ion Standard

Contract: 3-1A42HS1

Property Name: **www.mysite.com**

Create Property

Included Modules in Ion Premier or

- Access Control
- Adaptive Image Compression
- Advanced Offload
- Akamai Instant
- API Prioritization Cloudlet
- Application Load Balancer Cloudlet
- Audience Segmentation Cloudlet
- Cloud Monitor
- Content Targeting / EdgeScalability
- Edge Redirector Cloudlet
- Device Characterization

DEMO

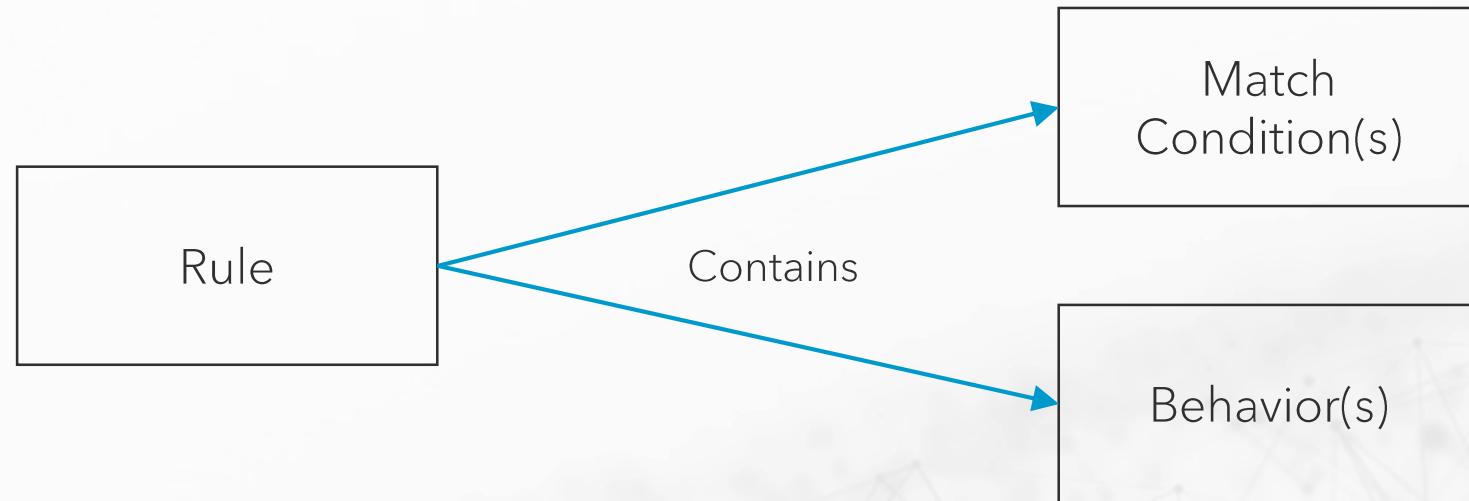
Property Manager



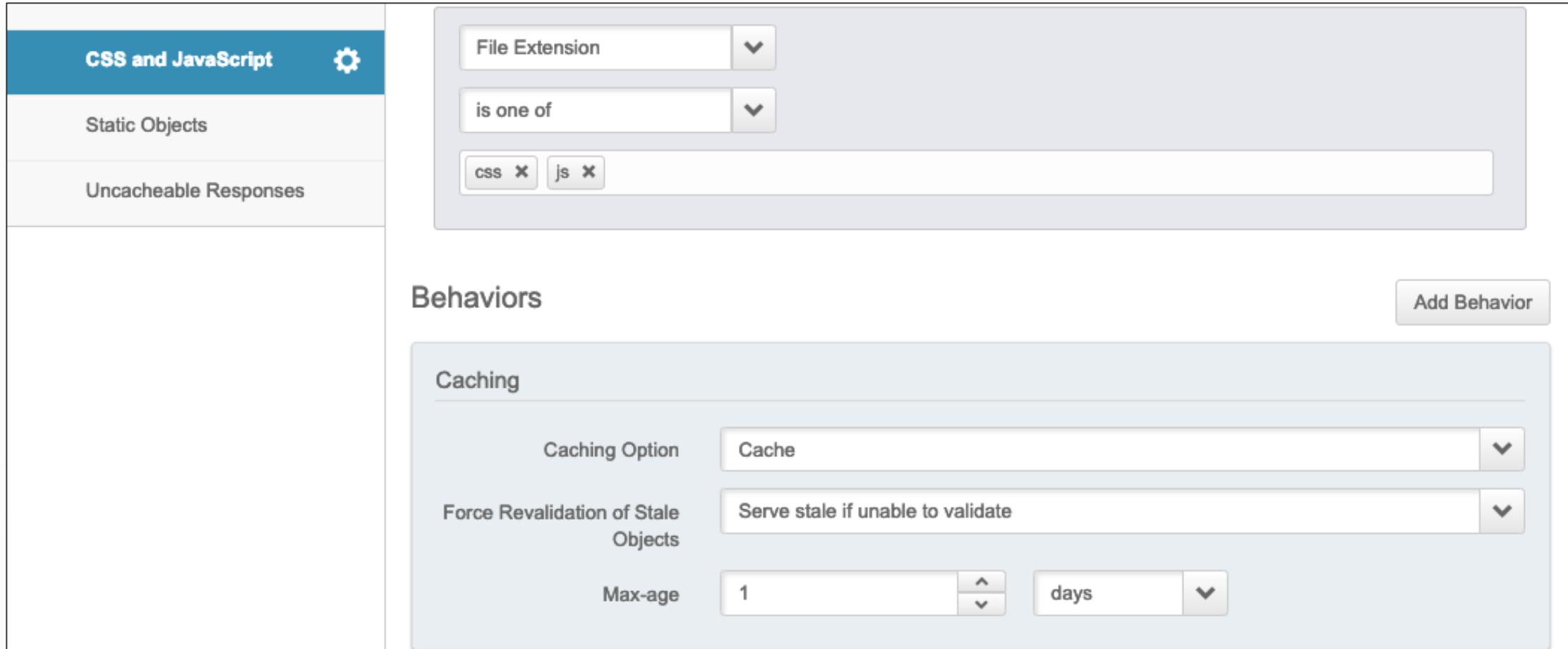
Rules, Match Conditions, Behaviors

In Property Manager, Rules = Match Condition(s) + Behavior(s)

The three fundamental objects for defining your functionality in your properties



Example of a Rule



The screenshot shows the Akamai Rule Editor interface. On the left, a sidebar menu includes 'CSS and JavaScript' (selected), 'Static Objects', and 'Uncacheable Responses'. The main panel displays a configuration for a rule:

- File Extension:** 'is one of' (css, js)
- Behaviors:** 'Add Behavior'
- Caching:**
 - Caching Option:** 'Cache'
 - Force Revalidation of Stale Objects:** 'Serve stale if unable to validate'
 - Max-age:** '1 days'

CP Codes

What is a CP Code?

- CP Codes (Content Provider Codes) are used in Akamai for traffic tracking, reporting, and billing purposes.
- All content delivered on the Akamai network must have a CP Code.
- CP Codes can also be used for purging content.
- They can be used to adjust the granularity of above mentioned features.
- Each content can have only one CP Code.

Setting CP Codes

CP Codes are associated to content via the Content Provider Code behavior.

Content Provider Code

Content Provider Code	315249 - webperformance 1	Create new...
-----------------------	---------------------------	---------------

Certain other functionalities allow you to specify a CP Code, for example the Image Manager behavior and NetStorage.

Q&A



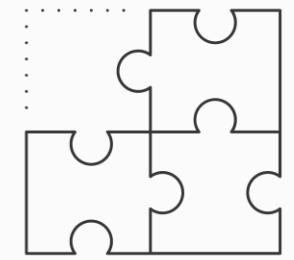
Summary

Fundamentals

- DNS Records
- Hostnames

Akamai Edge Hostnames

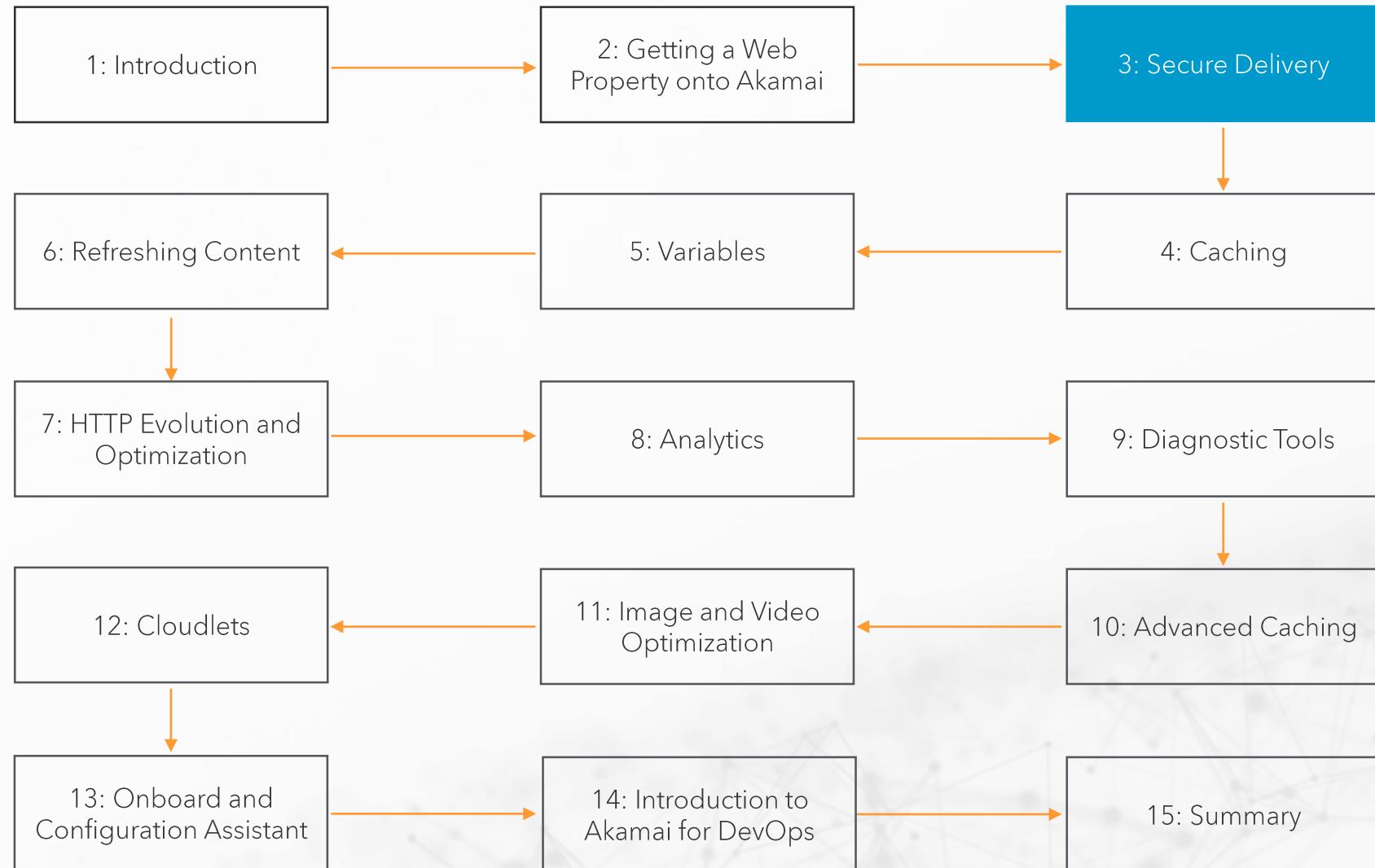
- edgesuite.net
- edgekey.net
- Staging Network



Akamai Control Center & Property Manager

- Akamai Control Center users and groups
- Property Manager is an app within Akamai Control Center
- Property Manager for configuring Akamai properties
- Rules, Match-Conditions, and Behaviors

Agenda



MODULE 3

Secure Delivery



Module Objectives

At the end of this module, you will be able to:

- Describe core TLS / SSL concepts.
- Describe certificate validation methods.
- Describe different certificate types.



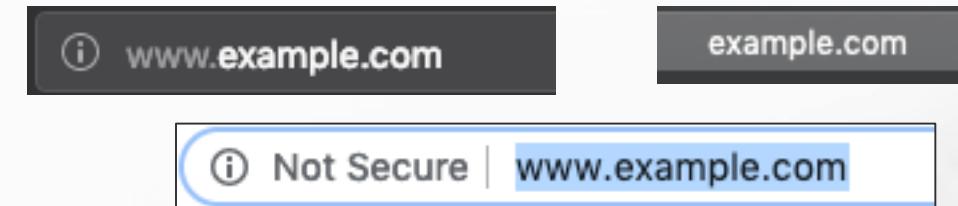
TLS - HTTPS

- New standards - TLS - Transport Layer Security - 1.0, 1.1, 1.2, 1.3
 - HTTP + SSL/TLS = HTTPS - Hypertext Transfer Protocol Secure
- * Old standards - SSL - Secure Socket Layer - 1.0, 2.0, 3.0
(v3 disabled on Akamai by default v1,v2 not available)



“TLS Everywhere” is a Reality

- Mandate for US government sites to be **secure**
- Google SEO rankings for **HTTPS**-enabled sites
- HTTP2 **requires HTTPS**
- Chrome, Firefox – new browser features for **HTTPS only**
- Non-HTTPS sites will be marked “**insecure**” by certain browsers



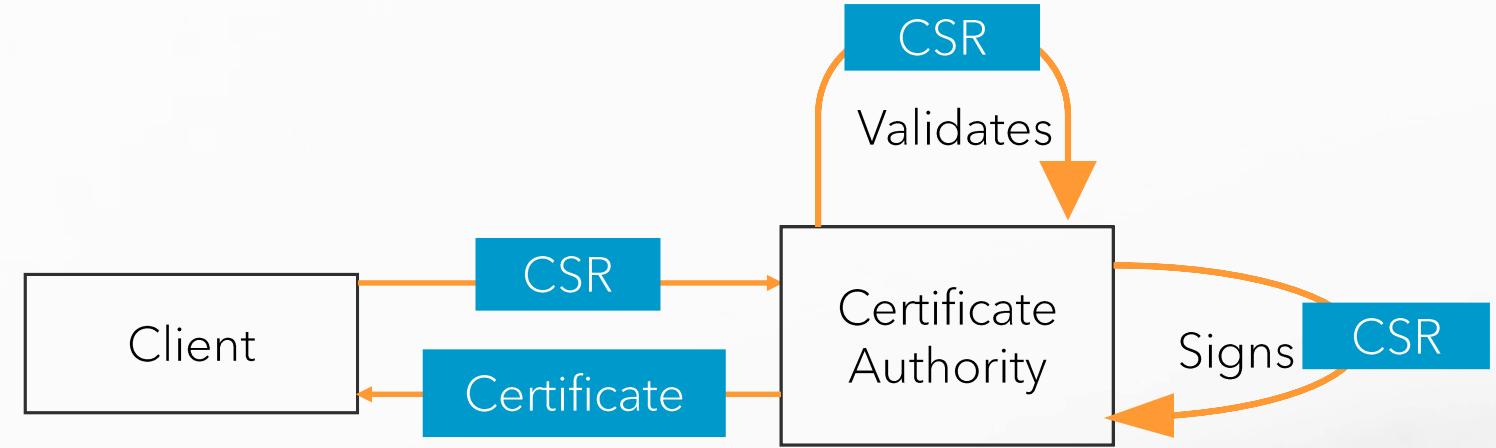
Anyone operating online needs a HTTPS strategy today.

Certificate Signing

The Certificate Authority Browser Forum and CASC:

- Governs the policies and procedures around the secure issuance of certificates.
- Sets baseline requirements: a subset of the requirements that a Certificate Authority must meet in order to issue digital certificates servers to be publicly trusted by browsers.

<https://cosecurity.org/ca-browser-forum-guidelines/>



Certificate Authority - Validation Types

Least validation, least expensive

Domain validation (DV)



Applicant must prove domain control only, no organization validation.

Organizational validation (OV)



Applicant must prove domain control and pass organizational vetting.

Extended validation (EV)



Applicant must prove domain control and pass extended organizational vetting.
CA verifies physical existence and integrity of organization.

Most extensive validation, most expensive

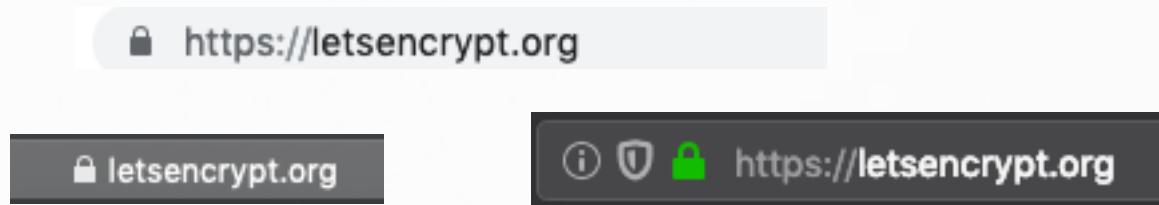
Client Trust

- It is inherited from trusted CAs root certificates.
- Depending on browser or system it can contain different values and might work with slight differences.
- Behavior can be modified by editing the trusted CA lists / stores.



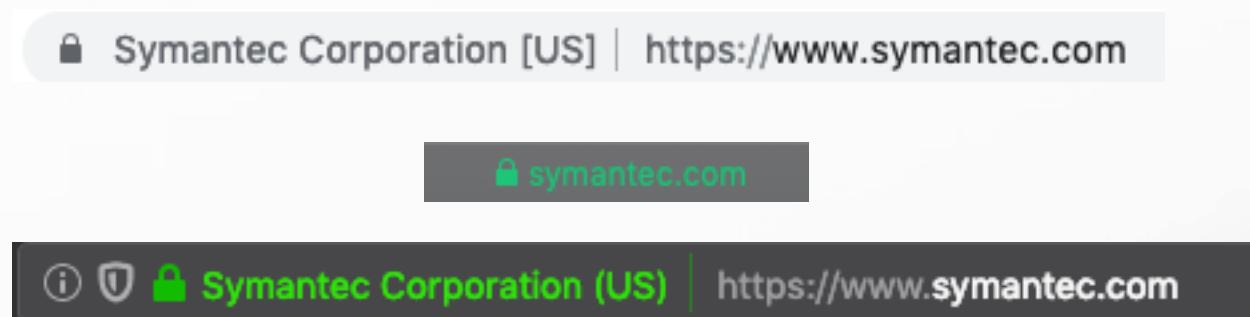
Browser Certificate Display - Trust

DV and OV certificate display on Chrome / Firefox / Safari



Visual cues vary between browsers

EV certificate display on Chrome / Firefox / Safari



Consistent visual cues between
browsers

- Green bar
- Organization name and country it resides

Certificate Types - SAN List

Singleton Certificate

Supports only a single fully qualified domain name

For example:

www.domain.com

Only a single entry allowed

Wildcard Certificate

Supports unlimited subdomains of a single wildcard domain identified on the certificate.

For example:

*.domain.com

support.domain.com, buy.domain.com

Multiple domains

SAN Certificate

Supports multiple domain names on a single Certificate

For example:

a.domain.com

b.newsite.com

Multiple domains

Wildcard SAN Certificate

Supports unlimited subdomains of a single wildcard domain identified on the certificate.

For example:

*.domain.com, *.otherdomain.com,

Support.domain.com, buy.domain.com

Multiple domains and/or wildcard entries

Certificate Methods - Summary

All certs are classified using both:

- Method of Validation (i.e. EV / OV / DV)
- Singleton, SAN, Wildcard, or Wildcard SAN

For example, an **EV SAN Cert** means that the Certificate Authority used the **Extended Validation** process when issuing the certificate, and the certificate itself supports a **finite number of fully qualified domain names**, listed on the certificate.

There is no difference in the quality of encryption used in the SSL / TLS protocols for one certificate type compared to another!

Akamai Networks - TLS Deployment

Akamai network certificate deployment types comparison:

	Standard TLS	Enhanced TLS	
		VIP	SNI - only
Certificate without SNI	a248.e.akamai.net, *.akamaihd.net, *.akamaized-staging.net, *.akamaihd-staging.net, *.akamaized.net	Customer certificate	None
Certificate with SNI	Customer certificate	Customer certificate	Customer certificate
Network serving traffic	EdgeSuite	EdgeKey	EdgeKey
Dedicated IP	No	Yes	No
Enhanced physical protection of servers	No	Yes	Yes

DEMO

Certificate Provisioning System (CPS)



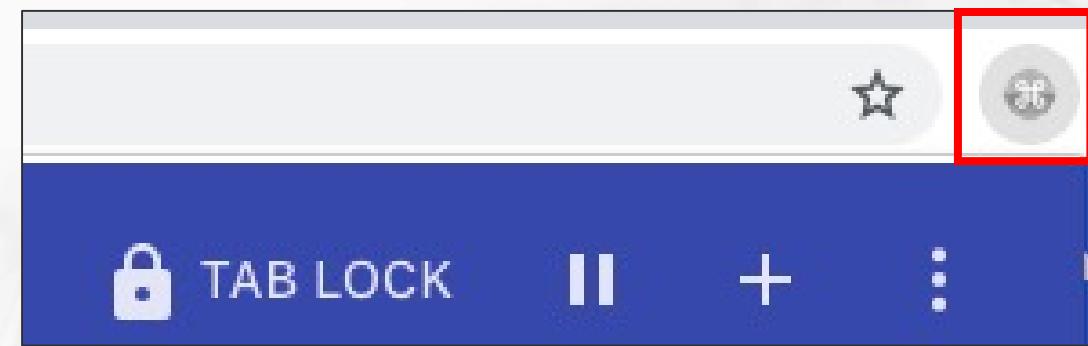
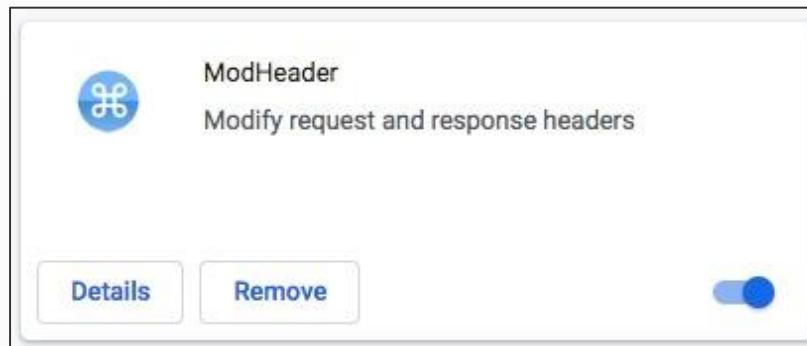
Tech Setup

Akamai Control Center Login

- <https://control.akamai.com/>
- User: wpX@akamaiuniversity.com

Software Requirements

- Install "Modify Headers" browser plugin in either Chrome or Firefox (google: "ModHeader" or "modify headers firefox")
- Some type of developer tools in Chrome or Firefox



LAB 1

Creating an Akamai Web Property



Q&A



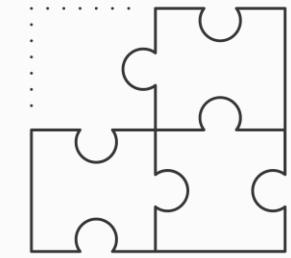
Summary

Certification Validation Levels

- Domain Validation
- Organizational Validation
- Extended Validation

Certificate Types

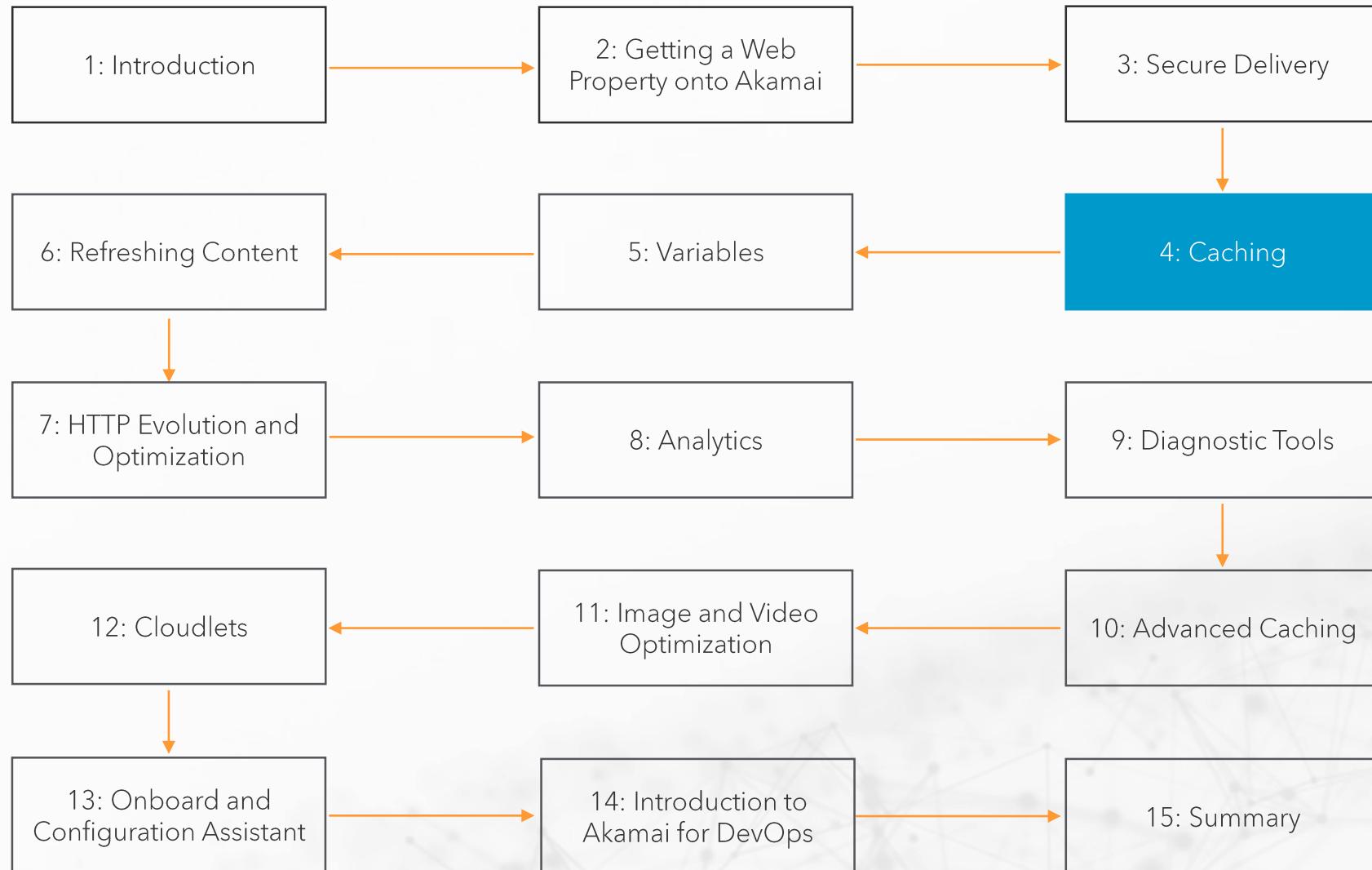
- Singleton
- SAN
- Wildcard
- Wildcard SAN



Certificate Provisioning System

- App within Akamai Control Center for creating certificates
- Certificate enrollment

Agenda



MODULE 4

Caching



Module Objectives

At the end of this module, you will understand:

- Caching at the Edge
- Caching at the Browser
- The Cache Key
- Using Akamai debug headers



Caching

Caching means storing resources for future reuse.

Caching content dramatically improves performance and provides origin offload.

There are two types of caching:

1. Caching at the Edge
2. Browser Caching

What should not be cached?

Almost anything can be cached:

static resources

- text files (css, jss, html, txt, xml)
- binary files (pdf fonts pictures)

dynamically generated content

- results of searches, listings
- API calls
- product pages

What cannot be cached:

Resource that is not reusable by others due to:

- uniqueness** - there is no benefit in the resource for other recipients.

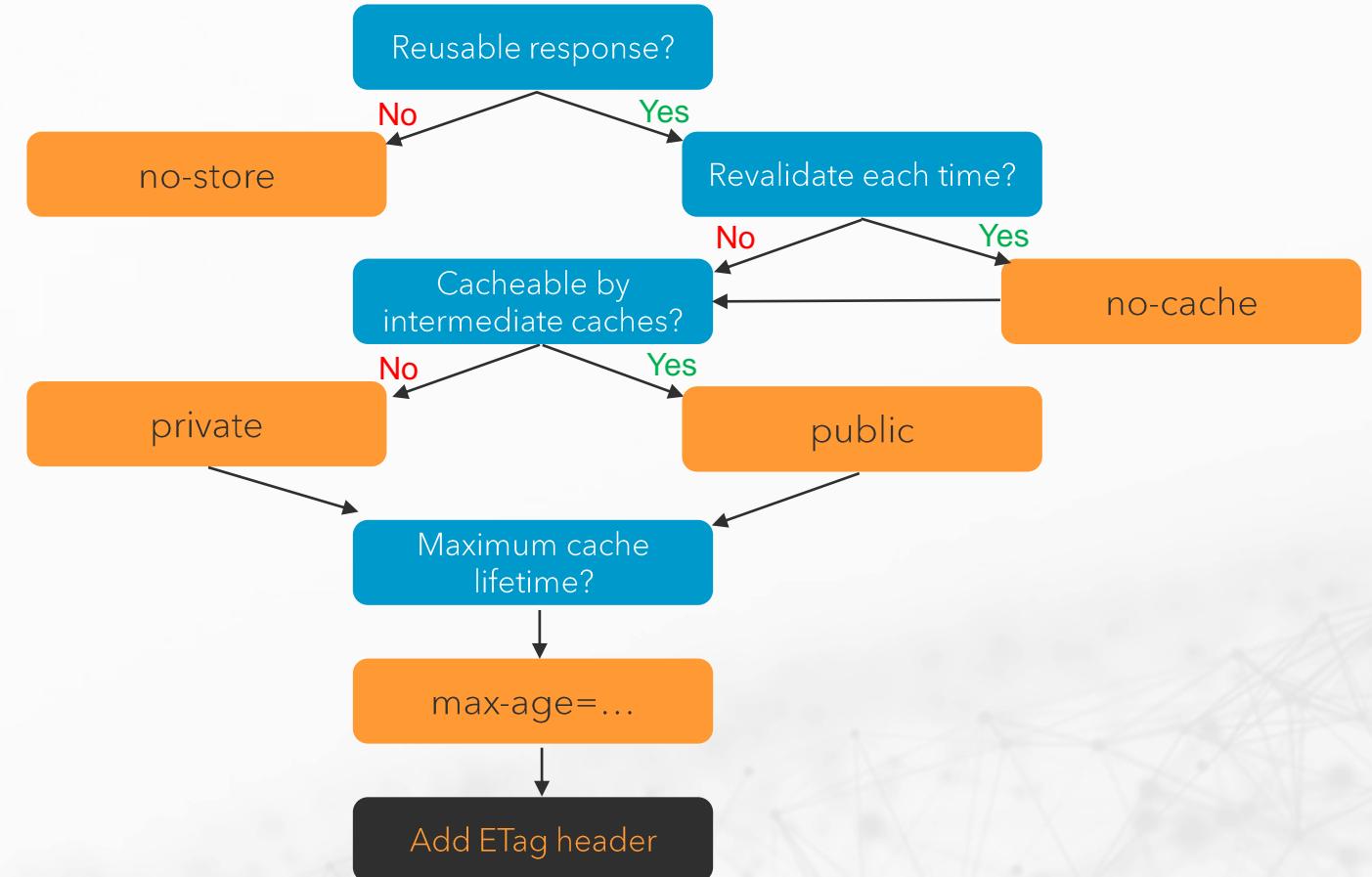
- type of information** - for example: persona/medical information caching will compromise security and privacy

- legal restrictions** - system must be complaint with regulations, which might prohibit caching.

Caching information

Information about caching can be conveyed in HTTP response headers.

Graph shows how to make a decision about setting Cache-Control header.



Caching at the Edge

Caching content on Akamai platform servers is defined by a dedicated behavior called "Caching".

Caching

Caching Option	Cache
Force Revalidation of Stale Objects	Serve stale if unable to validate
Max-age	0 <input type="button" value="^"/> <input type="button" value="▼"/> seconds <input type="button" value="^"/> <input type="button" value="▼"/>

Caching Behavior Options

Caching Behavior Options:

- Cache - cache (at the Edge)
- No Store - don't cache
- Bypass Cache - go to origin even if the object is in cache
- Honor Cache [...] - honor cache headers set at an origins

Caching	
Caching Option	Cache
Force Revalidation of Stale Objects	Cache
Max-age	No Store
	Bypass Cache
	Honor Origin Cache Control and Expires
	Honor Origin Cache Control
	Honor Origin Expires

Caching at the Edge

The Caching behavior should be encapsulated in an appropriate rule.

Example:

Rule
Name: CSS and JavaScript

Behavior
Cache for 1 Day TTL

What is this?

The screenshot shows the Akamai Edge Rules interface. A rule named "CSS and JavaScript" is selected. The interface is divided into sections: "Default Rule" (Performance, Compressible Objects, Offload), "Match Condition" (Criteria: Match Any, If: File Extension is one of css, js), "Behaviors" (Caching: Cache, Max-age: 1 day, Cache Refreshing: On, 90%), and "View Rule JSON" and "Add Match" buttons.

Match Condition
Match on CSS and JS types

Caching at the Browser

The “Downstream Cacheability” behavior is used for caching at the client / browser.

Downstream Cacheability

Caching Option	Allow caching
Cache Lifetime	(Default) Smaller value: origin header or remaining edge TTL
Send Headers	Send both Cache-Control and Expires
Mark as Private	<input type="checkbox"/> Off

Caching at the Browser

What are the options for caching?

Downstream Cacheability	
Caching Option	Allow caching
Cache Lifetime	Allow caching Allow caching, require revalidation (no-cache)
Send Headers	Don't allow caching (bust)
Mark as Private	Pass cacheability headers from origin Don't send cacheability headers; client browser defaults apply

Caching at the Browser

What Time To Live information will be send to client?

Downstream Cacheability

Caching Option	Allow caching
Cache Lifetime	(Default) Smaller value: origin header or remaining edge TTL
Send Headers	(Default) Smaller value: origin header or remaining edge TTL Greater value: origin header or remaining edge TTL
Mark as Private	Remaining edge TTL Full edge TTL (max-age) Fixed value Calculate Expires from origin Cache-Control

The Cache Key

Every object in the Akamai Cache has its own unique Cache Key.

For URL: <https://demo-basic.labs.akamaiuweb.com/product.php?productID=006>

`/demo-basic.labs.akamaiuweb.com/product.php?productID=006 vcd=1656`

Default* Cache Key contains:

- **Hostname (domain)** of the object: either the property hostname or the origin hostname
- **Path** - to the resource
- **Query String** - from the URL
- **Internal identifier** - shared on account level

*We will discuss the additional information that can be added to the cache key in a later module.

Akamai Debug Headers

Add the following Request Header to view Debug Headers:

Pragma: akamai-x-get-request-id, akamai-x-cache-on, akamai-x-cache-remote-on, akamai-x-check-cacheable, akamai-x-get-cache-key, akamai-x-get-extracted-values, akamai-x-get-nonces, akamai-x-get-ssl-client-session-id, akamai-x-get-true-cache-key, akamai-x-serial-no

This can be done using tools such as:

- Modify Headers (plugin for Chrome, Firefox)
- Charles HTTP Proxy
- cURL

```
curl -I -H 'Pragma: akamai-x-get-request-id, akamai-x-cache-on, akamai-x-cache-remote-on, akamai-x-check-cacheable, akamai-x-get-cache-key, akamai-x-get-extracted-values, akamai-x-get-nonces, akamai-x-get-ssl-client-session-id, akamai-x-get-true-cache-key, akamai-x-serial-no' https://demo-basic.labs.akamaiuweb.com/
```

Akamai Debug - Response Headers

Header Names:

- X-Cache
- X-Cache-Remote
- X-Cache-Key
- X-True-Cache-Key
- X-Check-Cacheable

DEMO

Akamai Debug Headers



Caching Challenges

What challenges exist with caching at the Edge?

What challenges exist with caching at the client?



LAB 2

Basic Caching



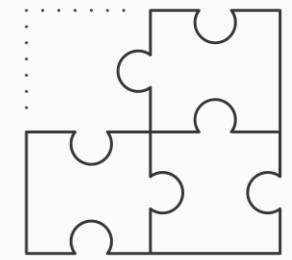
Q&A



Summary

Behaviors for Caching

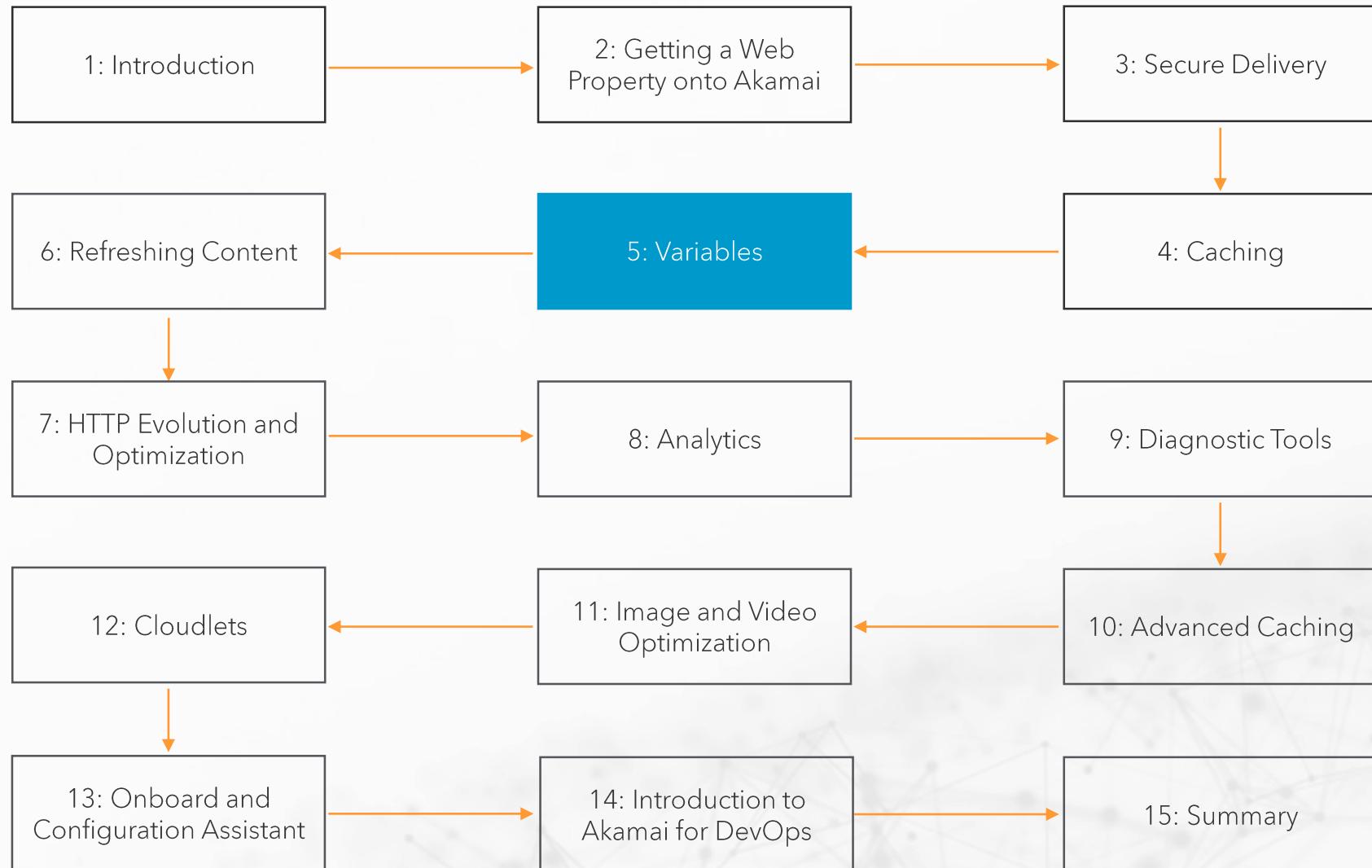
- Caching at the Edge with the Caching Behavior
- Caching downstream with the Downstream Cacheability Behavior



Caching Content

- Cache Key – host, path and query
- Different TTLs for different types of content
- Debug headers for testing & troubleshooting

Agenda



MODULE 5

Variables



Module Objectives

At the end of this module, you will understand:

- What Akamai Control Center Variables are.
- What you can do with Akamai Control Center Variables.
- Built-in and user-defined Akamai Control Center Variables.
- How to use Akamai Control Center Variables within Property Manager.



Introduction to Akamai Control Center Variables

Akamai Control Center Variable is a named value.

Akamai Control Center Variables allow you to:

- Create user-defined values.
- Use values stored in built-in variables.
- Reduce use of "advanced metadata".
- Implement complex logic on the Edge server.

Requirements / Pre-Requisites

Variables are supported in Property Manager.

Types of Akamai Control Center Variables

There are two types of Akamai Control Center Variables:

1. Built-In Variables
2. User-Defined Variables

Built-In Variables

- Set internally by Akamai Edge servers.
- Read-only - you can only reference them.
- Full reference of built-in variables:
<https://control.akamai.com/dl/property-manager/property-manager-help/GUID-ECC7523C-72DF-44D3-B56A-38632745EB70.html>



Variable Name	Short Name	Akamai Control Center Reference	Description
AK_CLIENT_IP	Client IP	{{builtin.AK_CLIENT_IP}}	Client IP address seen by the edge server. Can be overridden by the X-Forwarded-For and Akamai-Client-IP request headers.
AK_TLS_CIPHER_NAME	TLS Cipher Name	{{builtin.AK_TLS_CIPHER_NAME}}	For HTTPS and SPDY requests, this variable holds the name of the cipher used for the SSL connection. For HTTP requests, its value is UNKNOWN .
AK_HOST	Hostname	{{builtin.AK_HOST}}	Hostname of the incoming request.

User-Defined Variables

- Defined and set in configuration.
- Then can be set or modified using the **Set Variable** behavior.

Add a Behavior for this Rule

Available Behaviors (by Category)

All

Search available behaviors

Save POST DCA processing result

Scheduled Invalidation

Set Response Code

Set Response Cookie

Set Variable

Simulate Error Response Code

Set Variable

Set a variable which can be used in another rule in conjunction with a variable match.

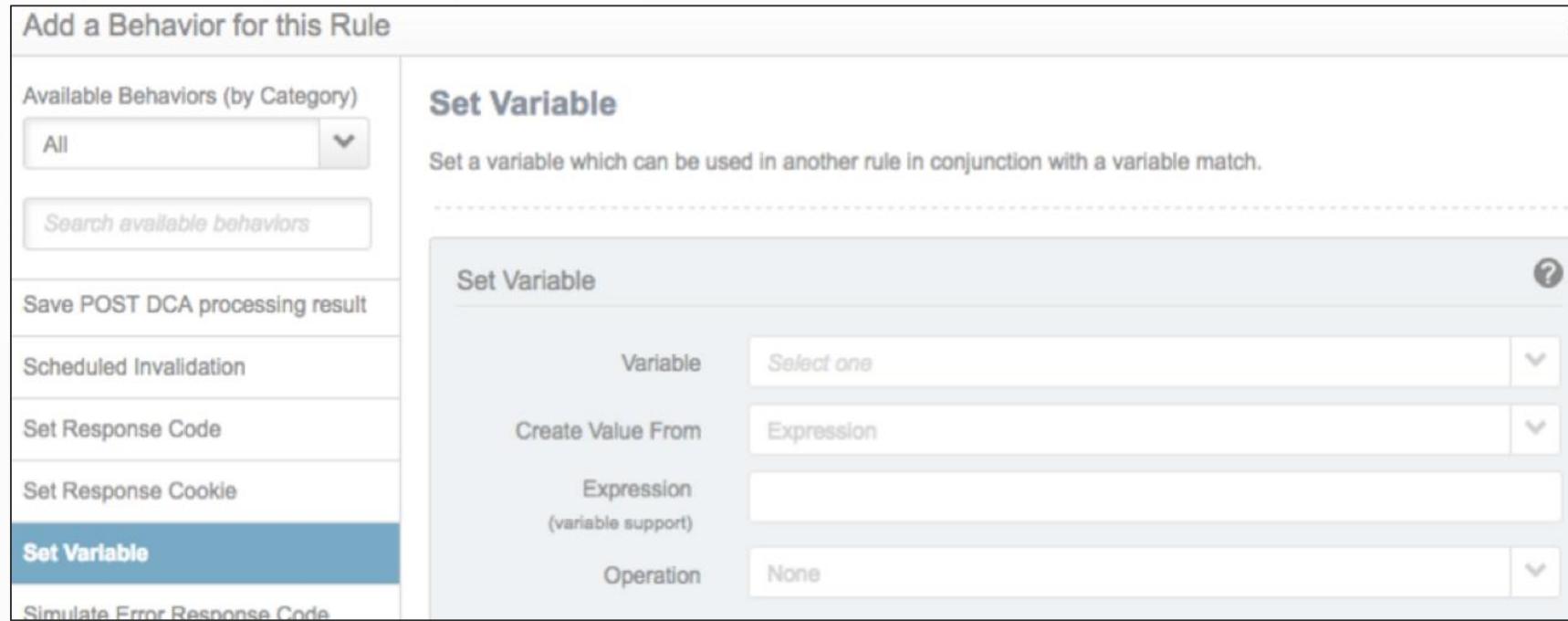
Set Variable

Variable: Select one

Create Value From: Expression

Expression (variable support):

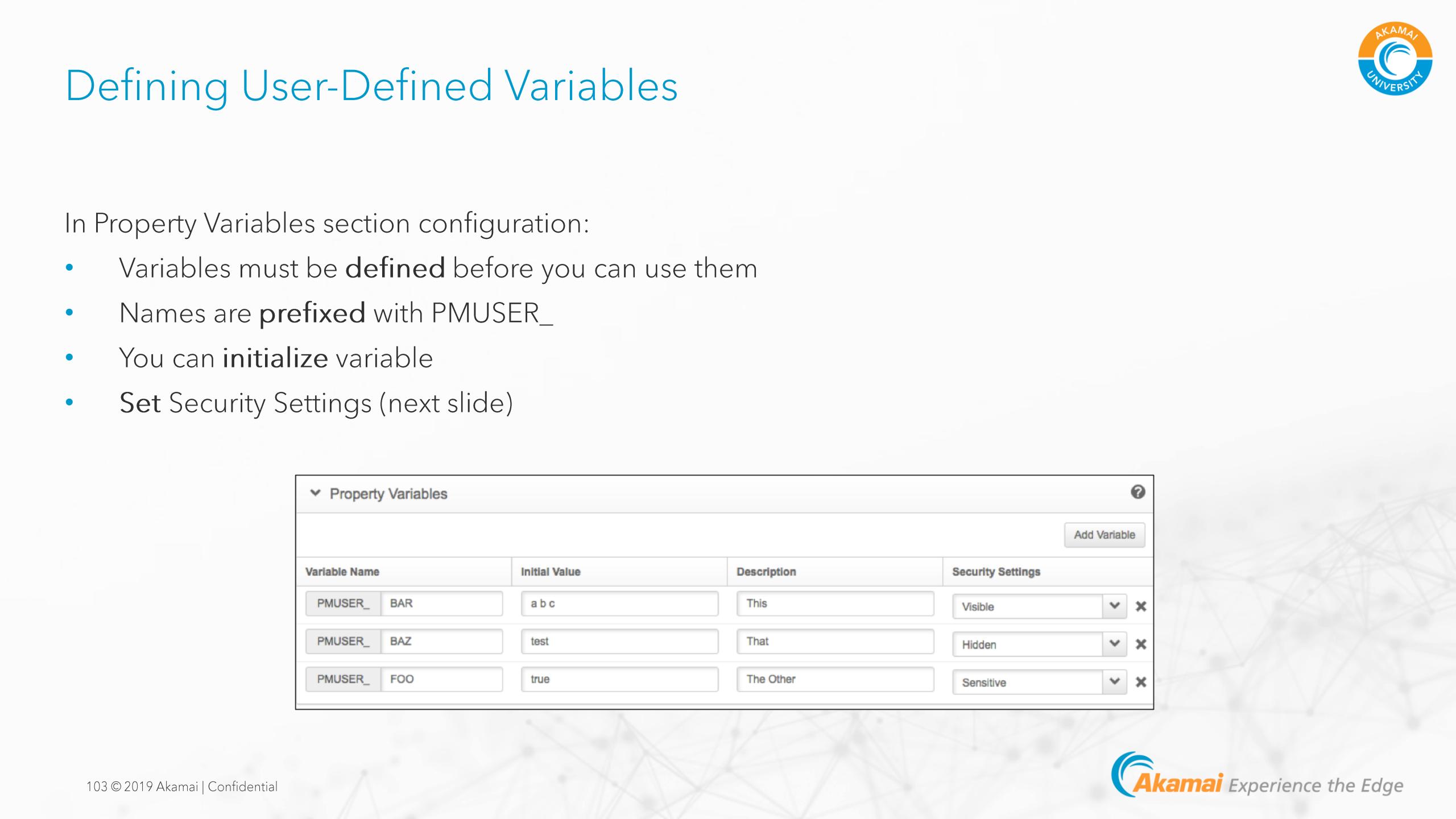
Operation: None



Defining User-Defined Variables

In Property Variables section configuration:

- Variables must be **defined** before you can use them
- Names are **prefixed** with PMUSER_
- You can **initialize** variable
- **Set Security Settings** (next slide)

A faint, light-gray network diagram consisting of numerous small, thin lines forming a complex web, serving as a background for the table.

Property Variables			
Variable Name	Initial Value	Description	Security Settings
PMUSER_BAR	a b c	This	Visible
PMUSER_BAZ	test	That	Hidden
PMUSER_FOO	true	The Other	Sensitive

Defining User-Defined Variables

The **Security Settings** control the access to the variable:

- **Visible** (default value) - will be returned to the client in the X-Akamai-Session-Info debug header
- **Hidden** - will not be returned in the X-Akamai-Session-Info debug header
- **Sensitive** - same like Hidden and:
 - Cannot be send to end-users through Set-Cookie or Modify Response Header behaviors.
 - Cannot be added to a custom logging field.
 - This attribute is inherited by any new variable created from this one.

Setting User-Defined Variables

The **Set Variable** behavior is used to set or modify the value of user-defined variables.

Add a Behavior for this Rule X

Available Behaviors (by Category) All ▼

[Search available behaviors](#)

Save POST DCA processing result

Scheduled Invalidation

Set Response Code

Set Response Cookie

Set Variable Set Variable

Simulate Error Response Code

Set Variable

Set a variable which can be used in another rule in conjunction with a variable match.

Set Variable

Variable	<input type="text" value="Select one"/> ▼
Create Value From	<input type="text" value="Expression"/> ▼
Expression (variable support)	<input type="text"/>
Operation	<input type="text" value="None"/> ▼

Set Variable Behavior

With the Set Variable behavior, you can assign a variable a value and manipulate it using one of three different methods (Create Value From):

- Expression
- Extract
- Generate

Set Variable

Variable	PMUSER_FOO
Create Value From	Expression
Expression (variable support)	<input type="text"/>
Operation	None

Create Value From - Expression

The Expression field contains the value assigned to the variable.
The value can include literal strings, as well as **builtin.** and **user.** variables.

There is code complete (type: {{)

Expression example: PMUSER_FOO

When hostname is: dev.site.com

Variable PMUSER_ORIGIN definition:
dev-origin.{{builtin.AK_DOMAIN}}

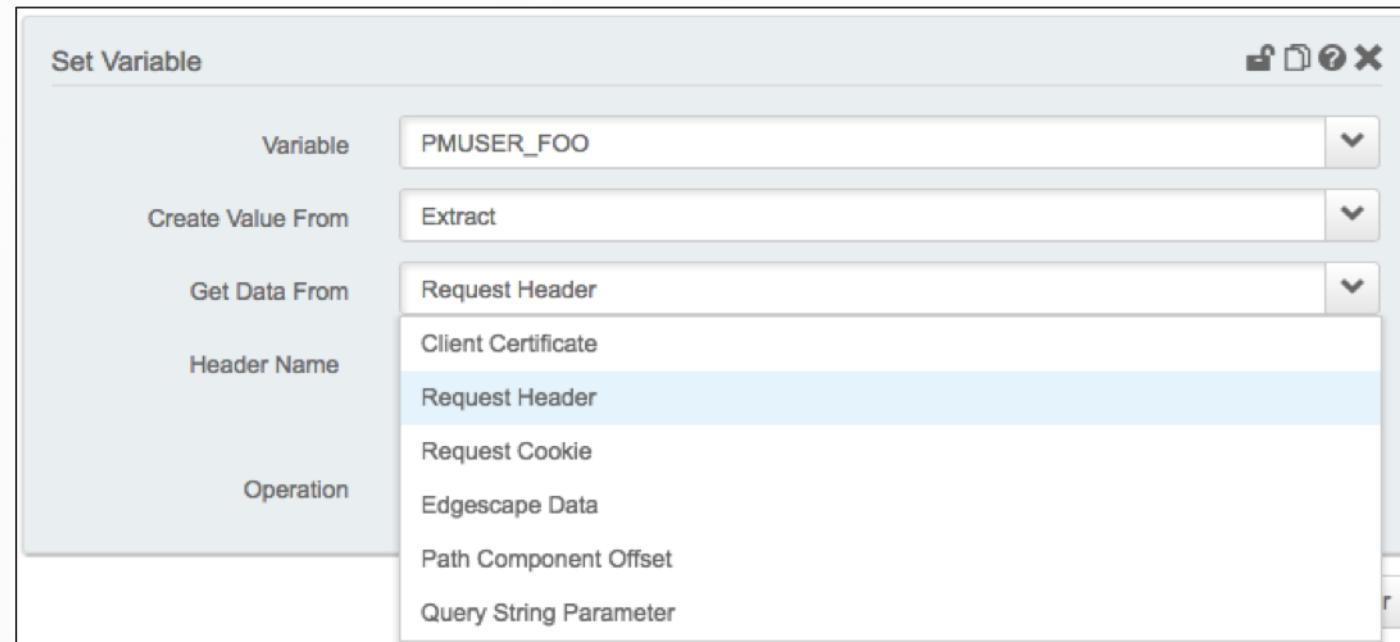
Variable value:
dev-origin.site.com

Set Variable

Variable	PMUSER_FOO
Create Value From	Expression
Expression (variable support)	<code>{{user.PMUSER_FOO}}fixedstring{{builtin.AK_PATH}}</code>
Operation	None

Create Value From - Extract

- Extracting data from the HTTP(S) request and assigning it to your variable
- The **Get Data From** field allows you to select from where to extract: headers, cookies, EdgeScape data, path components, query strings.



Create Value From - Generate

- For generating random numbers (decimal integers or hex)
- Specify the minimum and maximum value

Set Variable

Variable	PMUSER_FOO
Create Value From	Generate
Generator	Random Number
Enter the minimum numerical (variable support)	0
Enter the maximum numerical (variable support)	100
Operation	None

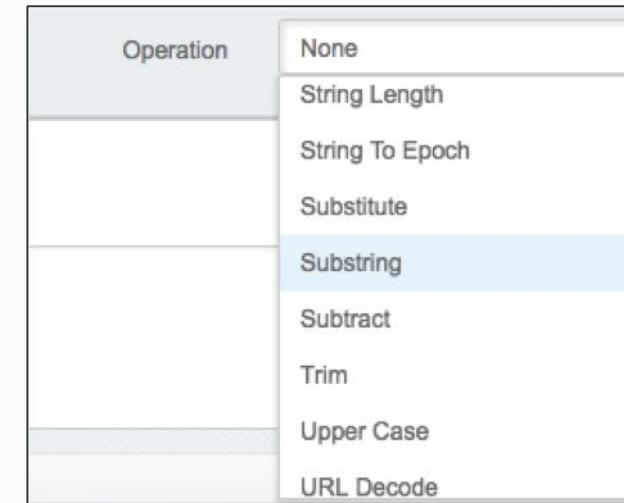
Create Value From - Operations

- Operations can be applied to the variable you are setting.
- Chain these operations by using more Set Variable behaviors.

Example: substring, trim, encoding, SHA256, upper/lower case, etc.

Set Variable

Variable	PMUSER_FOO
Create Value From	Expression
Expression (variable support)	<code>{{builtin.AK_URL}}</code>
Operation	URL Decode



Operations Specification:

<https://control.akamai.com/dl/property-manager/property-manager-help/GUID-2339B1C7-47EF-4C5D-B388-143C184BA2BD.html>

Create Value From - Operations

Additional Options

Some Operations require having additional options.

For example, if you select **Substring**, you must enter the start and end index.

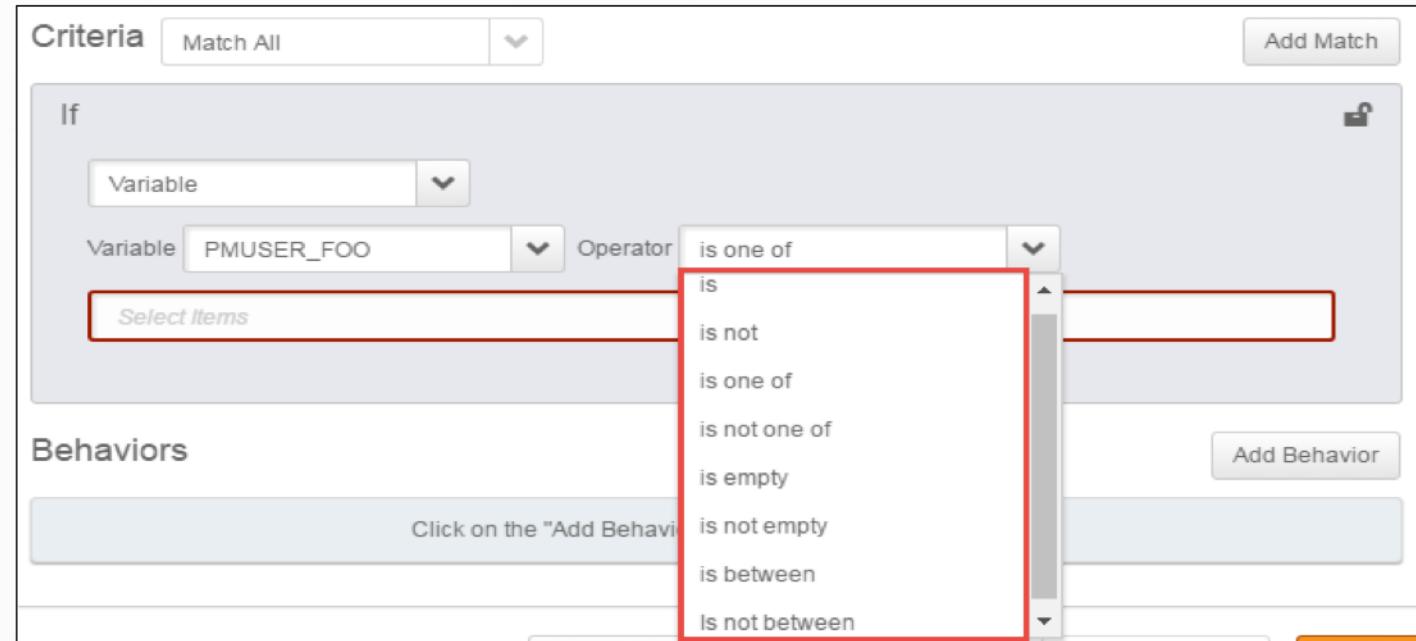
Set Variable

Variable	PMUSER_FOO
Create Value From	Extract
Get Data From	Request Header
Header Name	Accept-Language
Operation	Substring
Start Index (variable support)	0
End Index (variable support)	1

Matching on Variables

To create conditional logic use match on Variable.

It allows to build conditions in rules that are not included in standard matches.



Criteria Match All Add Match

If

Variable PMUSER_FOO Operator

Select Items

Behaviors

Click on the "Add Behavior"

is one of
is
is not
is one of
is not one of
is empty
is not empty
is between
Is not between

Variable Support

- Variables are supported in many behaviors
- Variable support in a form field is indicated by **(variable support)**

Example: Setting an outgoing request header with variable value

Modify Outgoing Request Header

Action	<input type="button" value="Add"/>
Select Header Name	<input type="button" value="Other..."/>
Custom Header Name (variable support)	Layout
Header Value (variable support)	<code>{{user.PMUSER_RWD_LAYOUT}}</code>

Variables – Customer Use Cases

When might you use Akamai Control Center Variables?



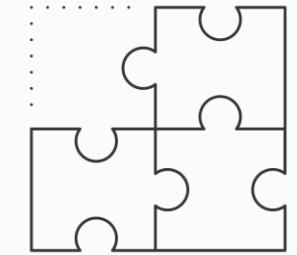
Q&A



Summary

Variables

- What Akamai Control Center Variables are
- Built-In Variables
- User-Defined Variables
- How variables are declared
- Different ways of setting Variables
- Matching on Variables
- Using Variables throughout Property Manager

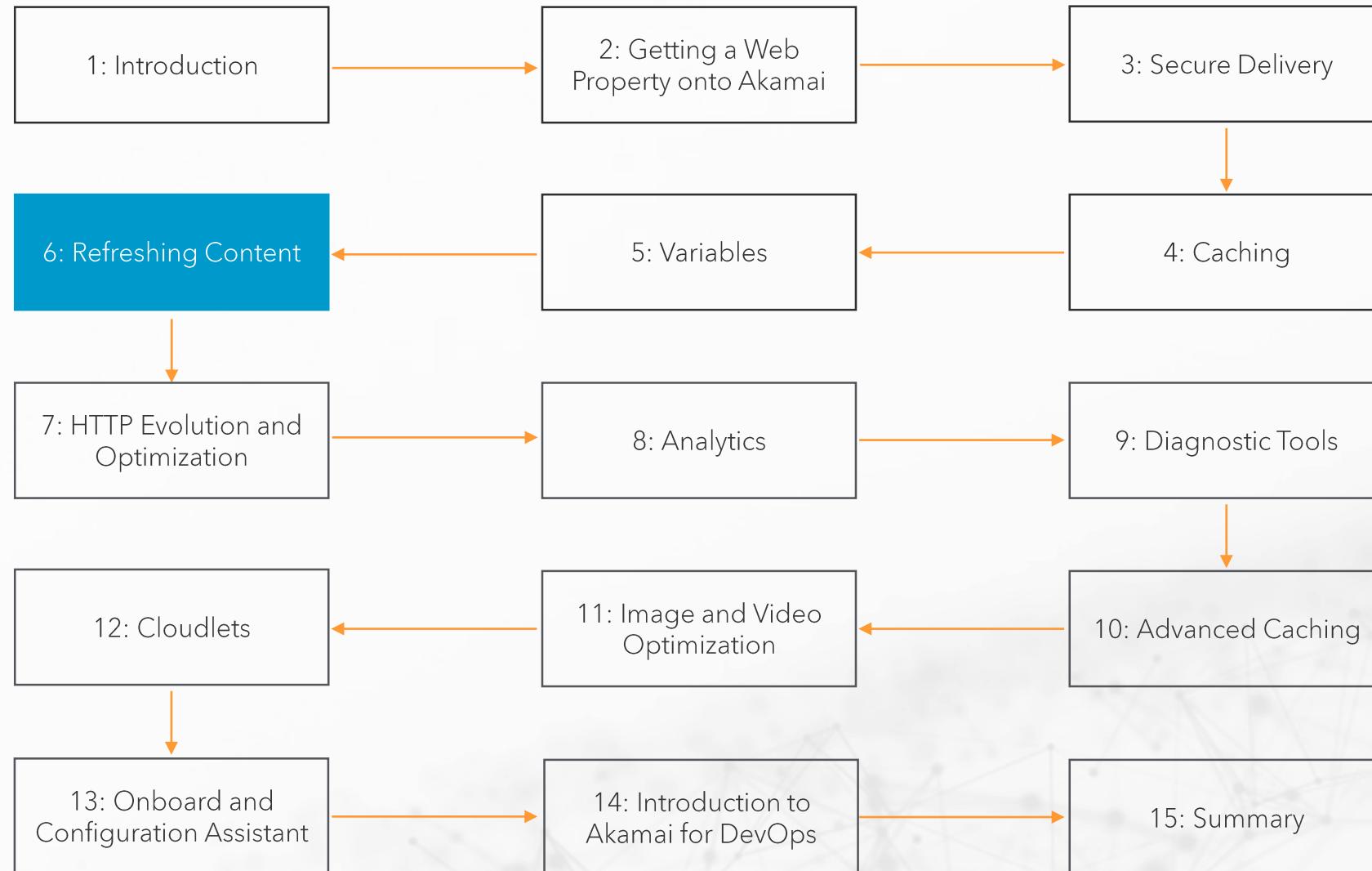


▼ Property Variables

Add Variable

Variable Name	Initial Value	Description	Security Settings
PMUSER_ BAR	a b c	This	Visible
PMUSER_ BAZ	test	That	Hidden
PMUSER_ FOO	true	The Other	Sensitive

Agenda



MODULE 6

Refreshing Content



Module Objectives

At the end of this module, you will understand:

- Options of removing objects from cache.
- Tools to perform purging.
- Differences between the tools.



Content Refreshing Methods

There are two types of Content Refresh:

Purge (Delete):

- Full-on deletes objects from the Akamai network
- Results in full object payload being fetched from Origin

Invalidate:

- Marks objects as expired in the cache, requiring re-validating before serving them
- Revalidation is done via If-Modified-Since request (make sure your origin supports these)
- Objects that have not changed can be revalidated by a mere header-only response from Origin

Whenever possible, use Invalidate rather than Purge!

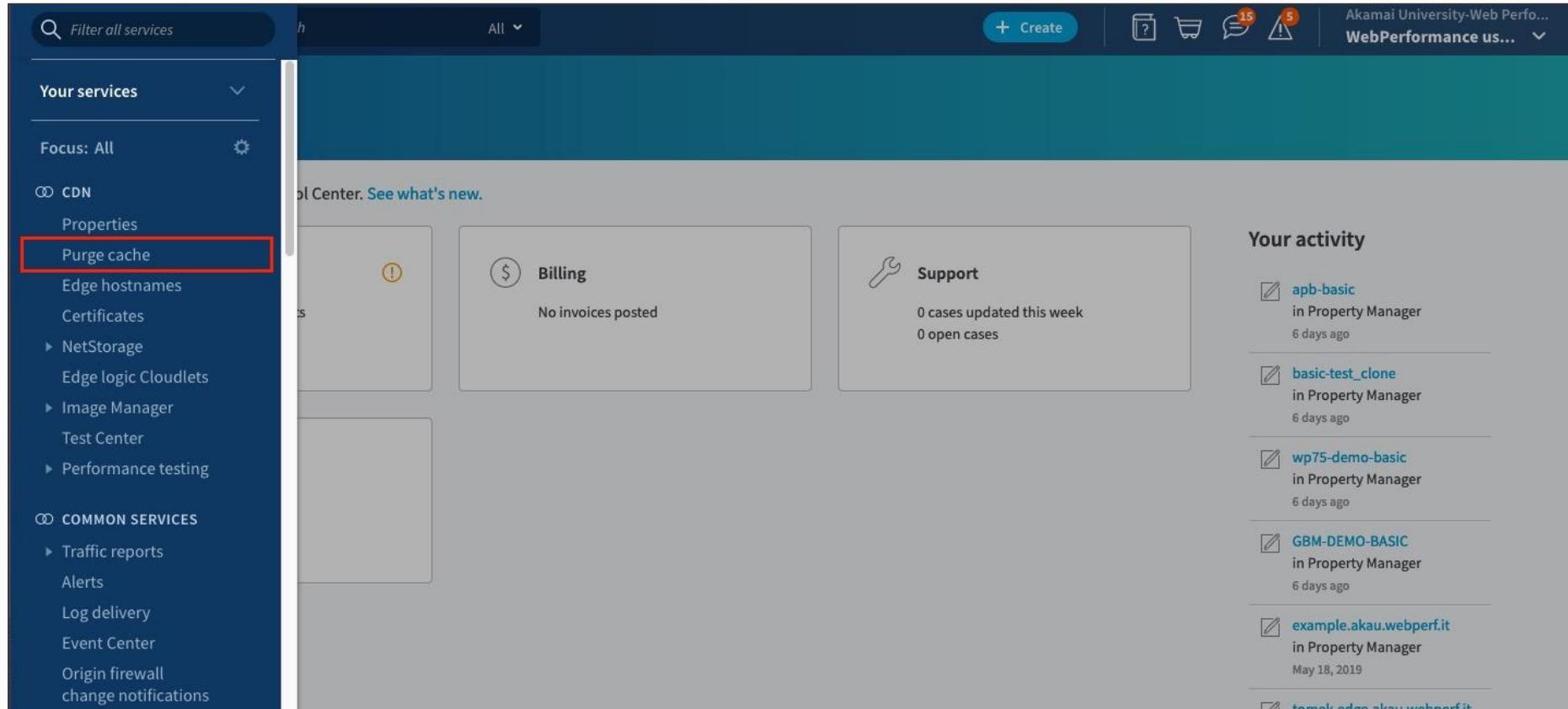
Content Refresh Tools

Tools for Refreshing Content:

- Fast Purge = Content Control Utility v3
- ECCU = Enhance Content Control Utility
- CCU = Content Control Utility v2 (will be deprecated soon)

Content Refresh Tools

Access to the Purge service can be done



The screenshot shows the Akamai Web Performance user interface. On the left, a sidebar menu titled 'Your services' is open, showing various service categories and specific items. The 'Purge cache' option under the 'CDN' section is highlighted with a red box. The main content area displays a dashboard with sections for 'Control Center', 'Billing', and 'Support'. On the right, a 'Your activity' sidebar lists recent updates in the Property Manager, each with a small icon and a timestamp (6 days ago).

- Focus: All**
- CDN**
 - Purge cache
 - Edge hostnames
 - Certificates
- COMMON SERVICES**
 - Traffic reports
 - Alerts
 - Log delivery
 - Event Center
 - Origin firewall
 - change notifications

Control Center. See what's new.

Billing
No invoices posted

Support
0 cases updated this week
0 open cases

Your activity

- apb-basic in Property Manager 6 days ago
- basic-test_clone in Property Manager 6 days ago
- wp75-demo-basic in Property Manager 6 days ago
- GBM-DEMO-BASIC in Property Manager 6 days ago
- example.akau.webperf.it in Property Manager May 18, 2019

Content Refresh Tools

Fast Purge

CCU

ECCU

Purge content by:

URLs
 CP Codes
 Cache Tags
 Cache-Keys / ARLs (Advanced)
 Directory and Extensions (Advanced) i

https://www.akamai.com/favicon.ico ?

File Upload ?
Przeglądaj...

Current Network ?
Production |||

Purge Method ?
Invalidate |||

Submit Purge

Content Refresh Tools

Create New ECCU Request

1 CREATE 2 CONFIRM

Request Name:

Properties: marcinj Selected Only 4

- alb.marcinj.bamse.pl
- esi.get.marcinj.bamse.pl
- esi.marcinj.bamse.pl
- esi.origin.marcinj.bamse.pl
- esi.post.marcinj.bamse.pl
- im.marcinj.bamse.pl
- marcinj.et
- marcinj.et.bamse.pl
- marcinj.et.edgesuite-staging.net
- marcinj.et.edgesuite.net

Filtered: 14 of 2,389 | Select 12 | Deselect 2

Emails:

Type: Directory ECCU File

Directory:

File Extensions:

Enhanced Content Control Utility

Help

New Request

Filter

Request ID Request Name Property Date Submitted Status Actions

There is no data based upon your criteria.

Confirm

Create request

Content Refresh Tools

Mechanisms	Fast Purge (CCUv3)	CCUv2	ECCU Use only when you need the advanced matching capabilities
Method of specifying content to Purge/Invalidate	<ul style="list-style-type: none"> • List of URLs, either in the UI, or file upload, or CP Code • Cache Tag 	<ul style="list-style-type: none"> • List of URLs, either in the UI, or file upload • By CP Code 	<ul style="list-style-type: none"> • By hostname • By directory or file extension • By uploading a list of URLs or directory/patch matches.
Support for Akamai Staging/Production Networks	Production and Staging	Production and Staging	Production Only
Supports for Purging and Invalidation	Invalidate and Purge	Invalidate and Purge	Supports only Invalidate
API-Support	Yes (REST) developer.akamai.com	Yes (REST) developer.akamai.com	Yes (SOAP) developer.akamai.com
Time it takes to Purge	< 5 seconds (Purge by URL) 4 minutes (Purge by CP Code)	5 minutes	Around 30-40 minutes
Availability	General Availability	Will get deprecated	General Availability

Content Refresh - {OPEN} APIs

Content Refresh can be done via {OPEN} APIs.

<https://developer.akamai.com/api/>

<https://github.com/akamai-open/api-kickstart>



API Fast Purge:

<https://developer.akamai.com/api/purge/ccu/overview.html>



POST

/ccu/v3/invalidate/url/production

Content-Type: application/json

{

 "objects": [
 "http://www.example.com/graphics/picture.gif",

 "http://www.site.example.com/graphics/picture.gif",

 "http://www.example1.com/documents/brochure.pdf"
]

}

DEMO

Refreshing Content



LAB 3

Refreshing Content



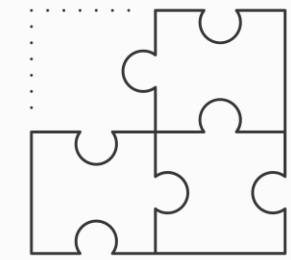
Q&A



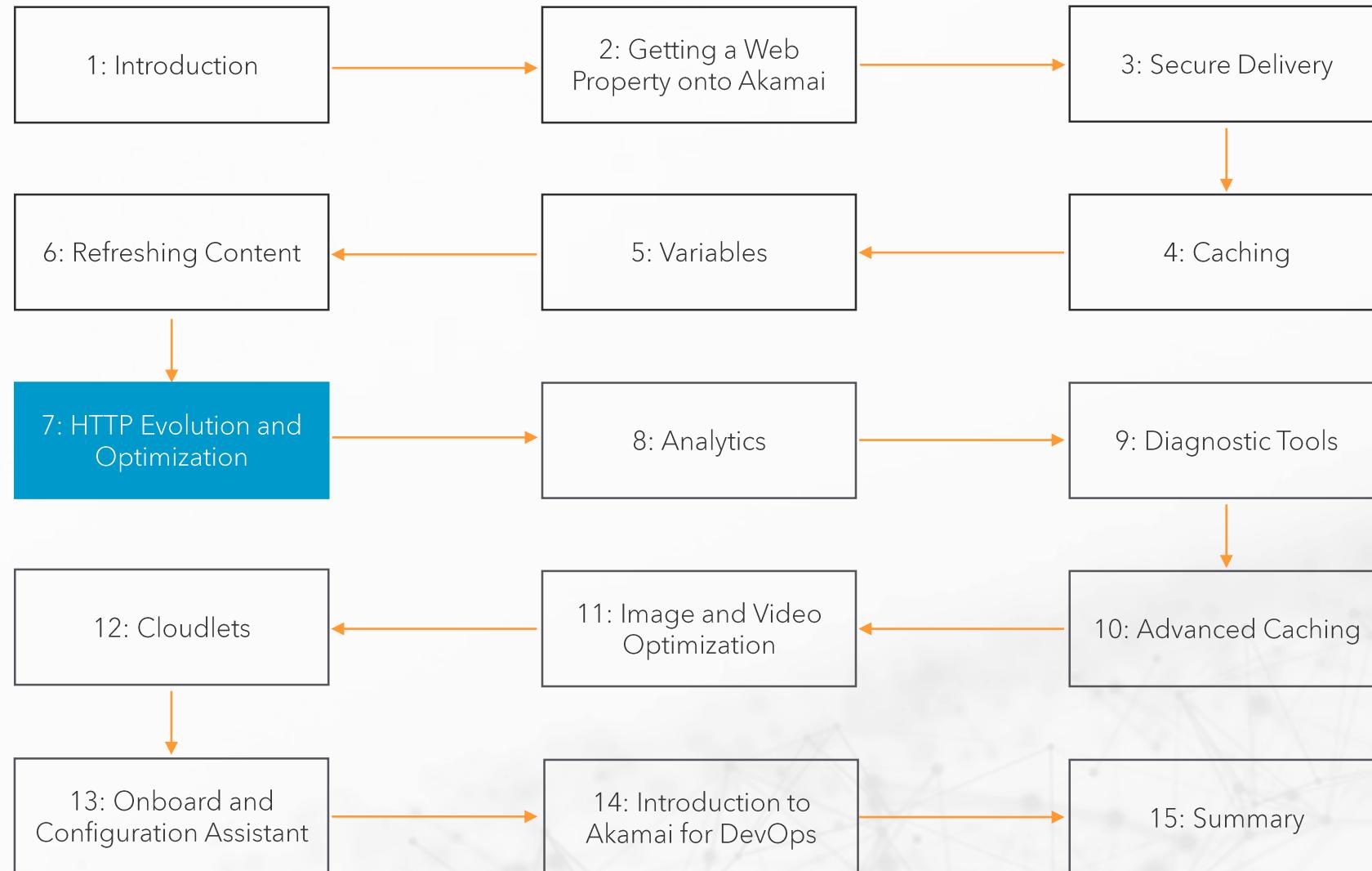
Summary

Content Refresh

- Invalidate vs Purge
- Fast Purge, CCU and ECCU
- Content Refresh APIs



Agenda



MODULE 7

HTTP Evolution and Optimization



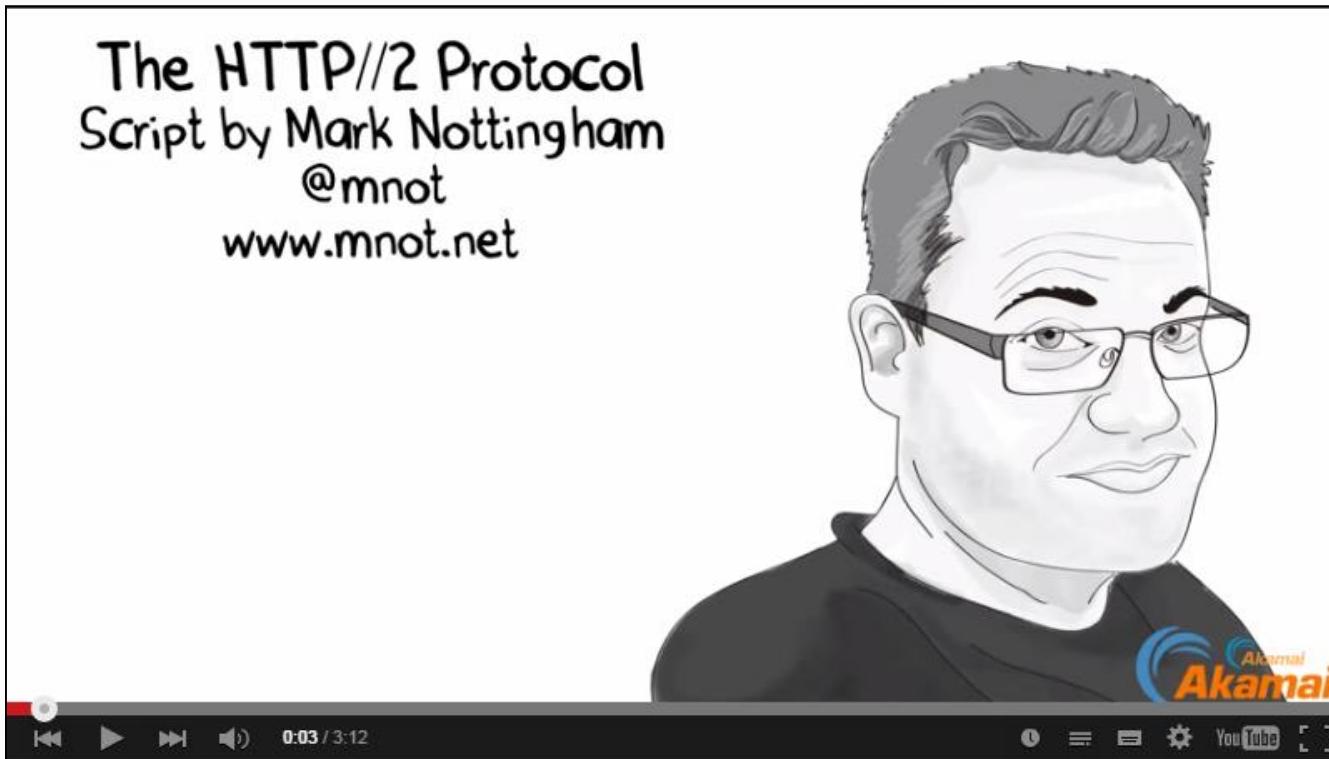
Module Objectives

At the end of this module, you will be able to:

- Describe how HTTP/2 enhances the standard HTTP protocol.
- Describe how HTTP/2 works.
- Describe how Akamai Adaptive Acceleration - Automated server push works.



HTTP 2.0



HTTP/s Origins

1991 HTTP/0.9		1999 HTTP/1.1		2015 HTTP/2
<ul style="list-style-type: none"> Client-server request-response protocol ASCII protocol 	<ul style="list-style-type: none"> Request may consist of multiple newline separated header fields Response object not limited to hypertext 	<ul style="list-style-type: none"> Allows multiple transactions over one persistent connection Supports chunked encoding 	<ul style="list-style-type: none"> Compresses request and response headers Allows multiple, multiplexed requests over a single connection 	<ul style="list-style-type: none"> Multiplexing & concurrency Stream dependencies Header compression
1996 HTTP/1.0		2009 SPDY		

Why HTTP/S Is Important

HTTP/2 attempts to solve many of the shortcomings and inflexibilities of HTTP/1.1:

- Improves performance by 5-25% over secure HTTP1.1
- Simplifies content creation (no hacks needed to speed load times)
- Drives adoption of HTTPS (improves security)
- Resource savings:
 - Less connections open on web servers
 - Bandwidth savings due to header compression

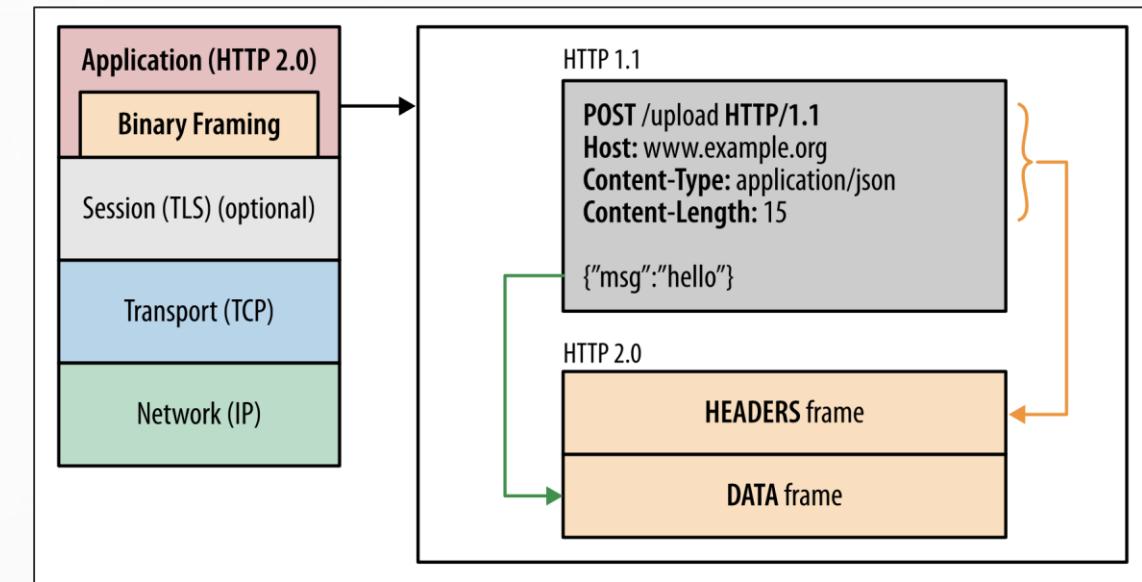
Akamai and HTTP/2

Akamai was driving HTTP/2! Here's how:

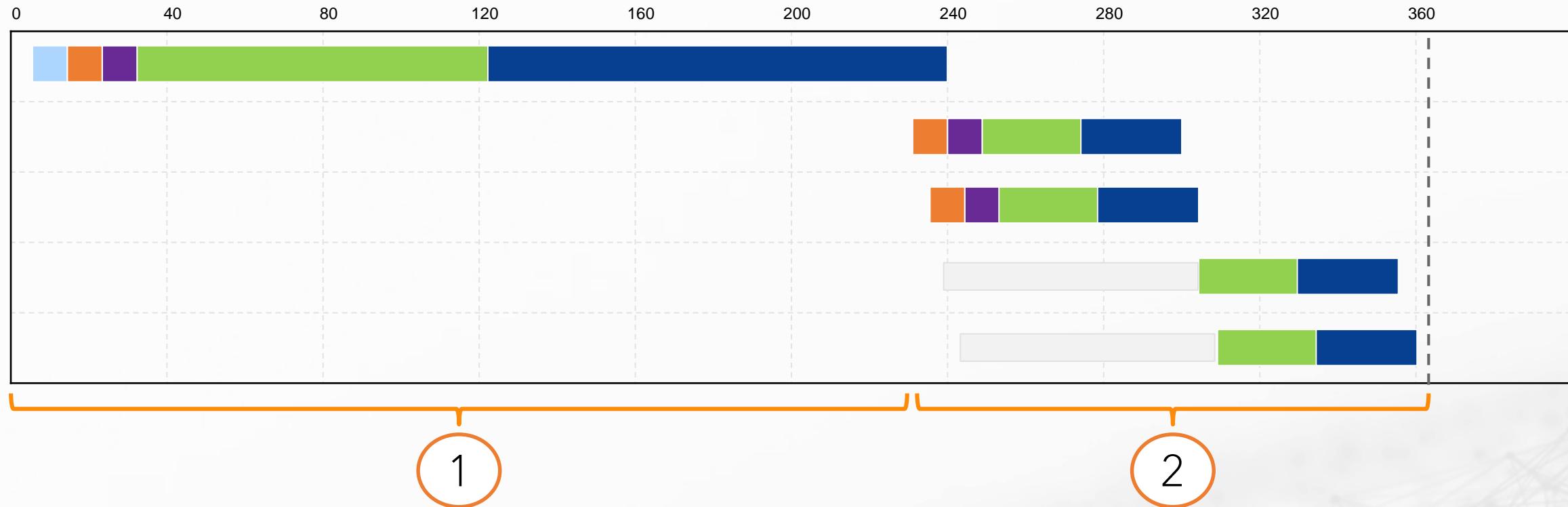
- Akamai was an active contributor to the HTTP/2 specification.
- Mark Nottingham of Akamai is chair of the IETF HTTP Working group.
- Stephen Ludin contributed development tools for SPDY and HTTP/2.
- Akamai was an early implementer of HTTP/2 (and SPDY).

HTTP/2 in a Nutshell

- Based on SPDY.
- It is a binary code, not textual.
- It uses a single TCP connection to the server.
- Frames are composed of request and response messages.
- Each request belongs to a stream.
- Streams are independent and can be prioritized by the client.
- Headers can be compressed.

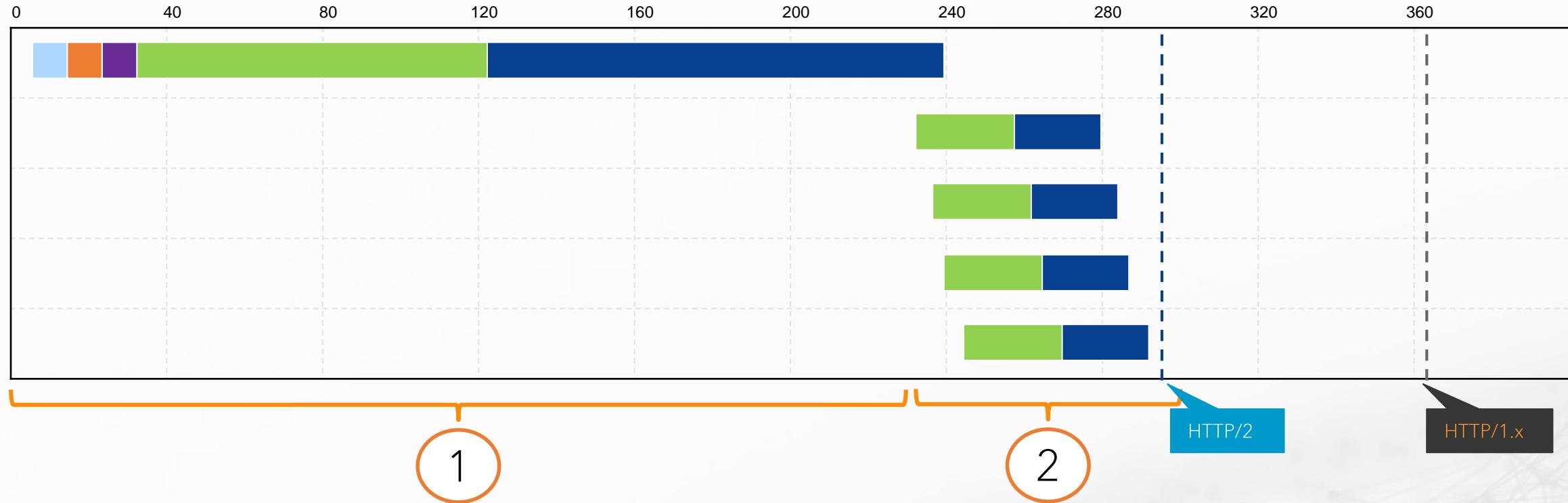


HTTP/1.x



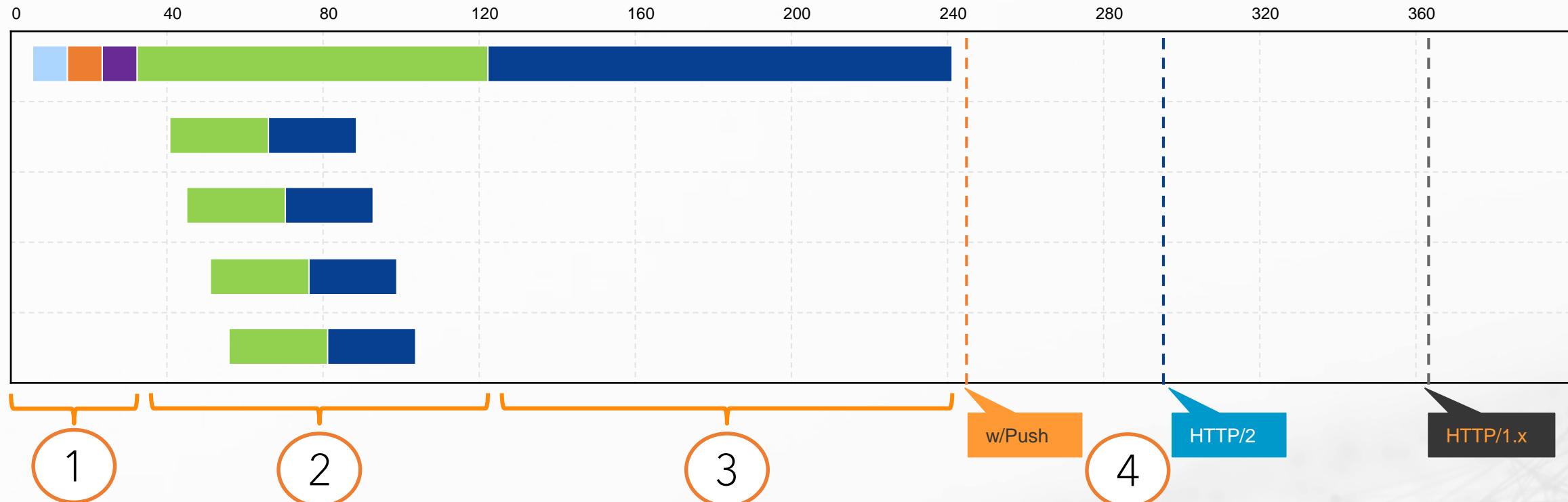
1. Main page loads first
2. Resources load afterwards – parallelism limited by TCP connections

HTTP/2



1. Main page loads first
2. Resources load afterwards – multiplexed and compressed headers from HTTP/2

HTTP/2 with Server Push

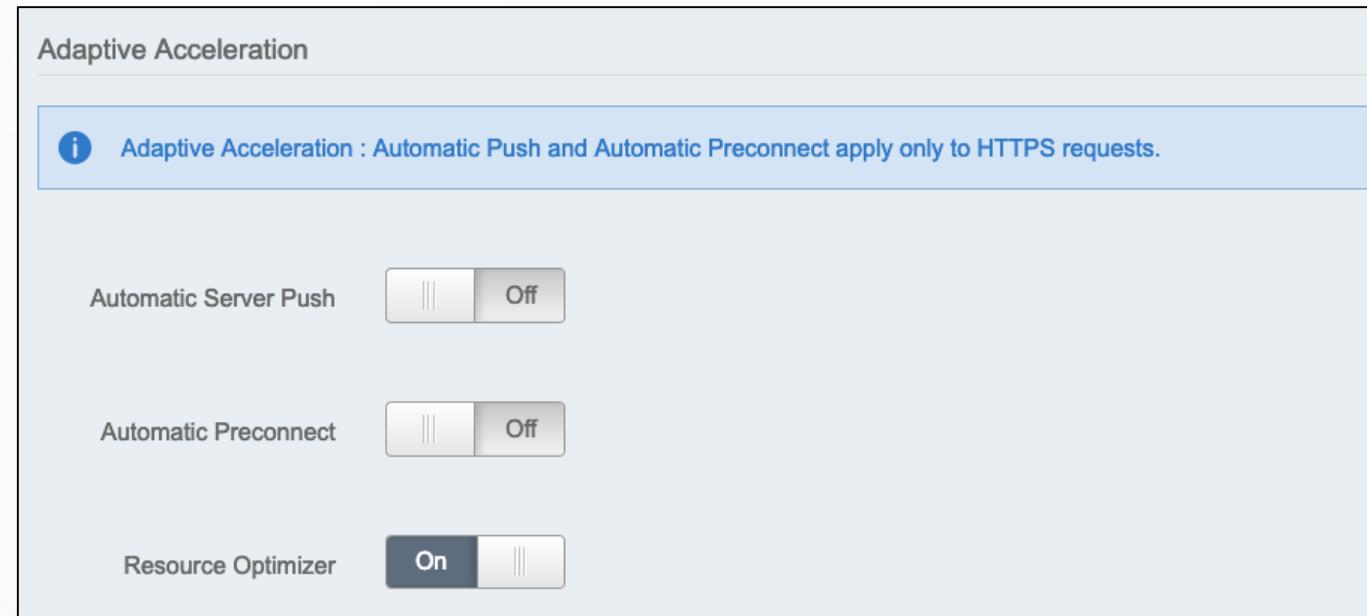


1. Main page is requested
2. Resources are pushed immediately (while the main page response is pending)
3. Main page is sent to the user uninterrupted
4. Resources are already at the browser

Adaptive Acceleration - Push & Preconnect

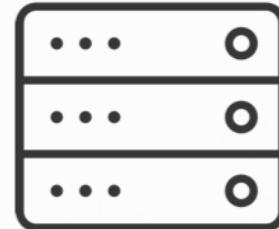
Adaptive Acceleration is a feature that allows to leverage server push functionality, preconnect and additional resource compression.

Adaptive Acceleration currently utilizes Resource Timing data from mPulse to anticipate which sub-resources will be requested before the browser actually requests them.



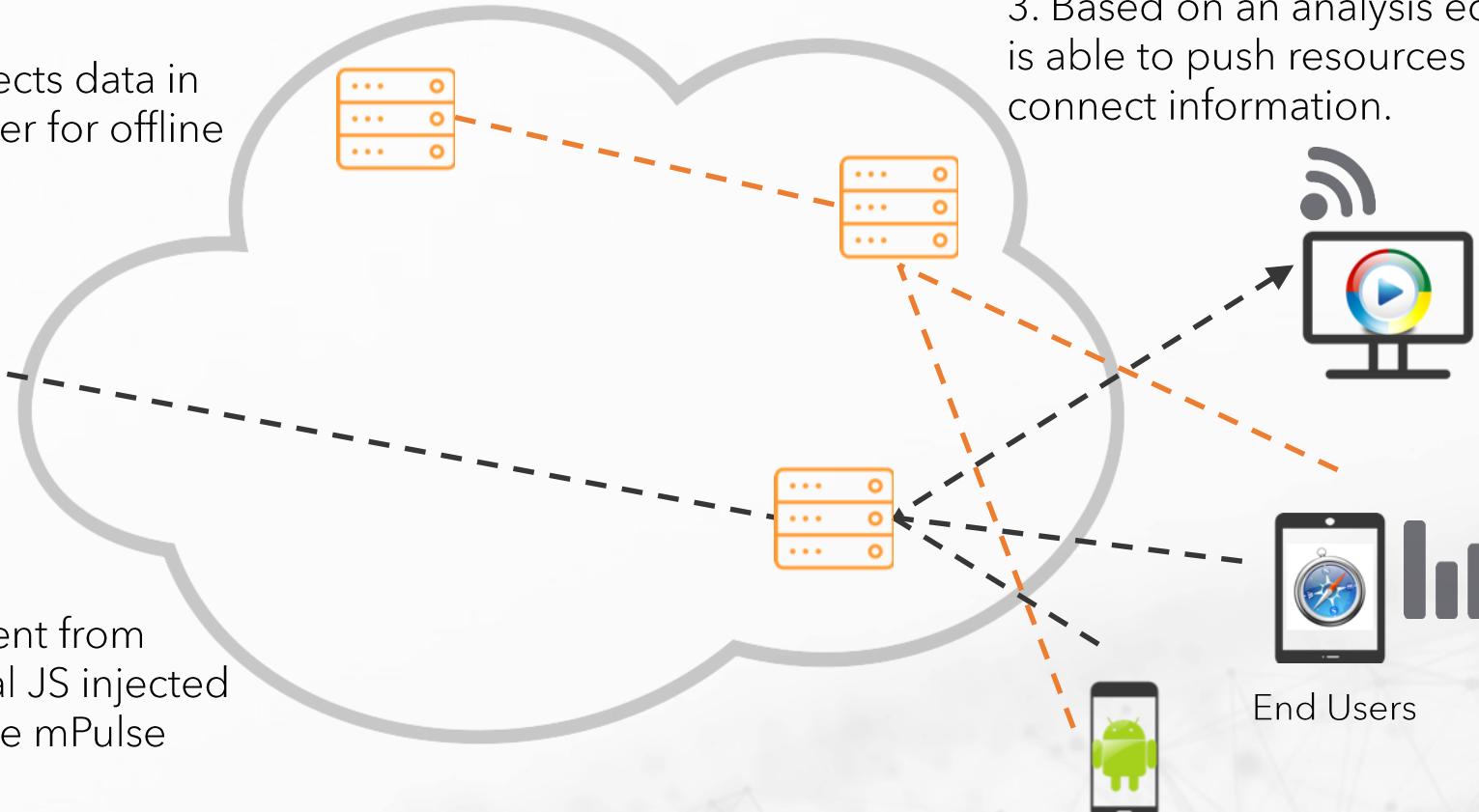
Adaptive Acceleration - Push & Preconnect

2. JavaScript code injects data in beacons to edge server for offline analysis.



Content Origin

1. Client requests content from origin. There is a special JS injected (origin/edge) to provide mPulse data.



3. Based on an analysis edge server is able to push resources or give pre-connect information.

Adaptive Acceleration - Resource Optimizer

Resource Optimizer Features

- Optimizes CSS, JS, and font files objects using Brotli and Zopfli compression.
- Creates compressed derivative objects offline (similar to Image Manager).
- Serves the best derivative based on HTTP/1 vs HTTP/2 and browser information inferred via the user-agent, accepted-encoding request header.

Resource Optimizer Extended Compatibility

Enable

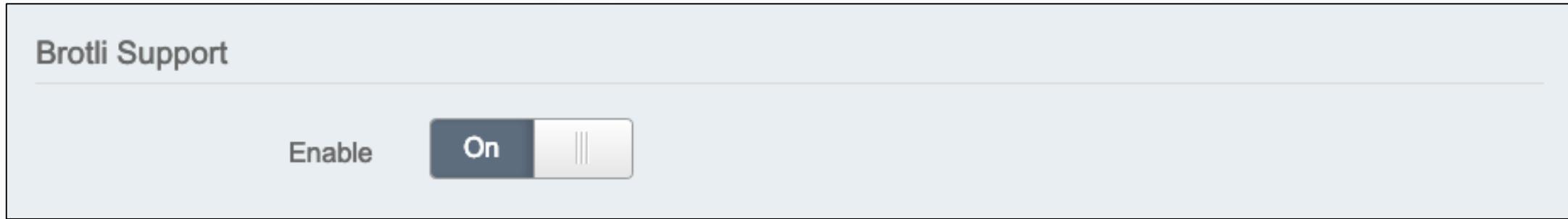
Latest Features

Adaptive Acceleration - Resource Optimizer

Compression Type	'content-encoding' Response Header	Support
Brotli <ul style="list-style-type: none"> alternate compression format to gzip - provides better compression 	content-encoding: br	<ul style="list-style-type: none"> http://caniuse.com/#feat=brotli HTTPS Only with RO
Zopfli <ul style="list-style-type: none"> gzip compatible, better compression - decoded the same as ordinarily gzip 	content-encoding: gzip	<ul style="list-style-type: none"> All browser that support gzip HTTP Only with RO

Brotli Support from Origin

Brotli Support behavior allows the CDN to return Brotli-compressed assets from your origin and cache them on edge servers.



What's Next?

HTTP/3 is coming - an HTTP over a QUIC transport protocol.

- Goal: 0-RTT Connection Establishment
- No Head Of Line Blocking
- QUIC Encryption
- QUIC Forward error correction
- Connection Migration
- Pluggable Congestion Control

Experimental protocol, not yet standardized.

LAB 4

HTTP/2 and Adaptive Acceleration



Q&A



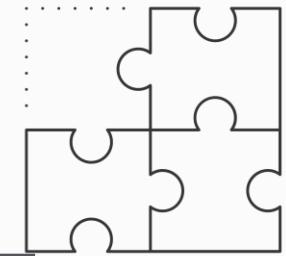
Summary

HTTP/2

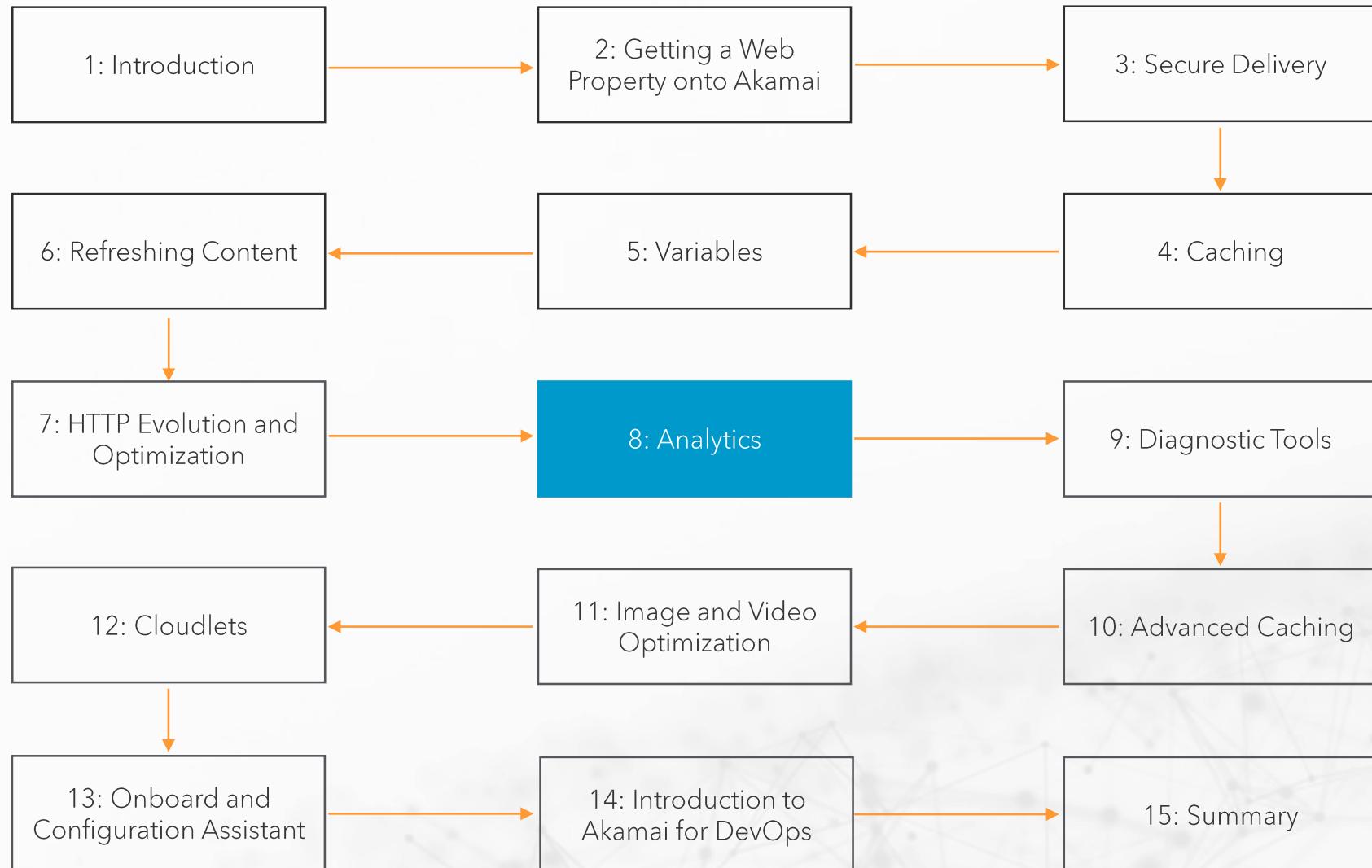
- HTTP/2 helps mitigate challenges in HTTP/1.1
- Akamai has been an early adopter of HTTP/2
- All Akamai core delivery products (Ion Standard, Ion Premier, DSA, etc) support HTTP/2
- Adaptive Acceleration's Server Push functionality pushes resources to the client before they are requested

Adaptive Acceleration

- Server Push helps to intelligently put required resources to the client
- Resource Optimizer, helps reduce the size of CSS and JS objects using more advanced compression types
- Brotli Support from Origin, allows the CDN to serve brotli-compressed resources to requesting browsers



Agenda



MODULE 8

Analytics



Module Objectives

At the end of this module, you will be able to:

- List the intelligent features that enable you to derive insights.
- Describe Akamai Control Center Reporting.
- Describe different logging solutions.



User Insights

Features that enable you to derive insights include:

- mPulse - Real User Monitoring solution
- Akamai Control Center Reports
- Log Delivery Service
- Cloud Monitor



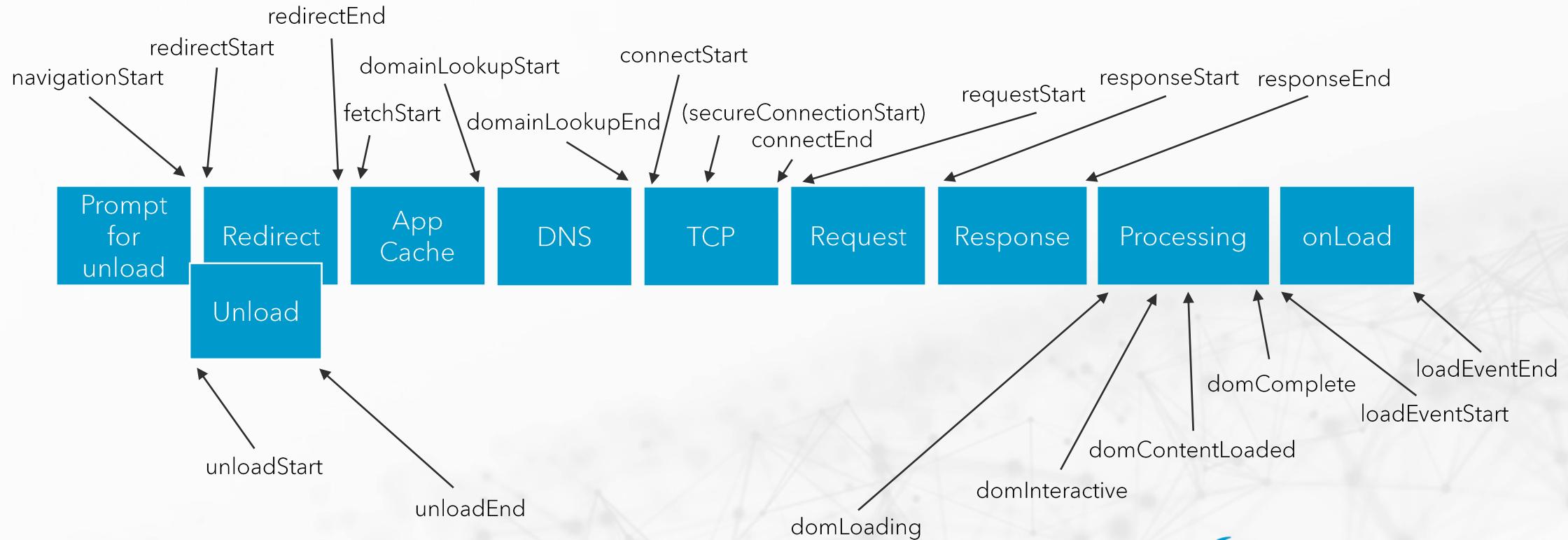
Navigation Timing Interface

Navigation Timing is a JavaScript API for accurately measuring performance on the web. It is accessed in JavaScript via the `window.performance.timing` object:

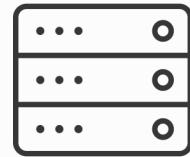
```
interface PerformanceTiming {  
    readonly attribute unsigned long long navigationStart;  
    readonly attribute unsigned long long unloadEventStart;  
    readonly attribute unsigned long long unloadEventEnd;  
    readonly attribute unsigned long long redirectStart;  
    readonly attribute unsigned long long redirectEnd;  
    readonly attribute unsigned long long fetchStart;  
    readonly attribute unsigned long long domainLookupStart;  
    readonly attribute unsigned long long domainLookupEnd;  
    readonly attribute unsigned long long connectStart;  
    readonly attribute unsigned long long connectEnd;  
    readonly attribute unsigned long long secureConnectionStart;  
    readonly attribute unsigned long long requestStart;  
    readonly attribute unsigned long long responseStart;  
    readonly attribute unsigned long long responseEnd;  
    readonly attribute unsigned long long domLoading;  
    readonly attribute unsigned long long domInteractive;  
    readonly attribute unsigned long long domContentLoadedEventStart;  
    readonly attribute unsigned long long domContentLoadedEventEnd;  
    readonly attribute unsigned long long domComplete;  
    readonly attribute unsigned long long loadEventStart;  
    readonly attribute unsigned long long loadEventEnd;  
};
```

Navigation Timing Interface: Processing Model

Each performance attribute shows the time of a navigation event or page load event, measured in milliseconds since midnight of January 1, 1970 (UTC).



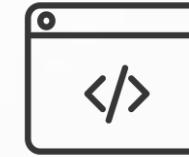
Perceived Performance



Server



Network



Browser



User

Synthetic

APM

Traditional RUM

RUM w/ Perceived Performance

Do you really know when the page is ready for use?

Are deferred elements blocking interactivity?

Impacted by CPU limitations on mobile?

Backend Time

Network Time

Page Load Time

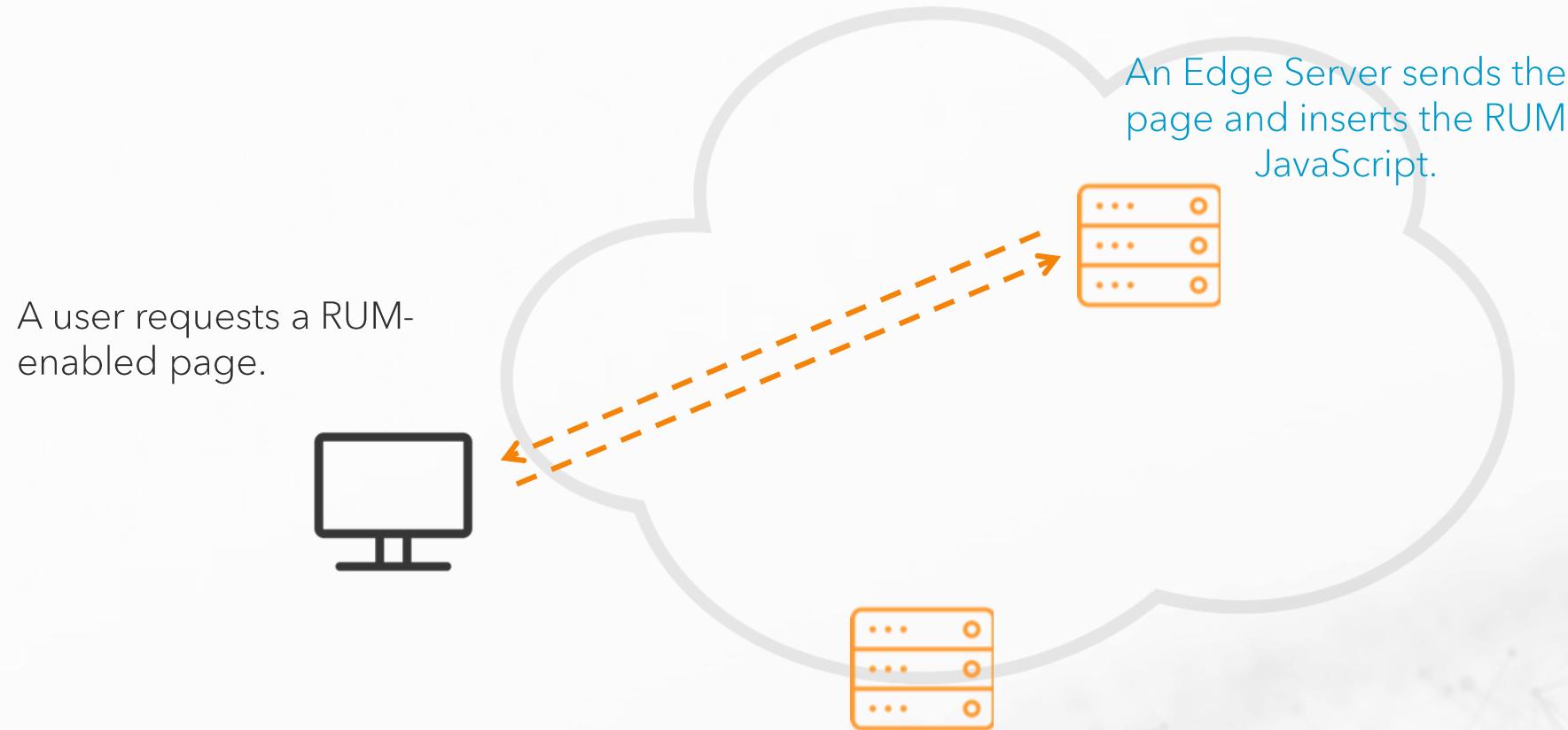
Interactivity

How Real User Monitoring (RUM) Works

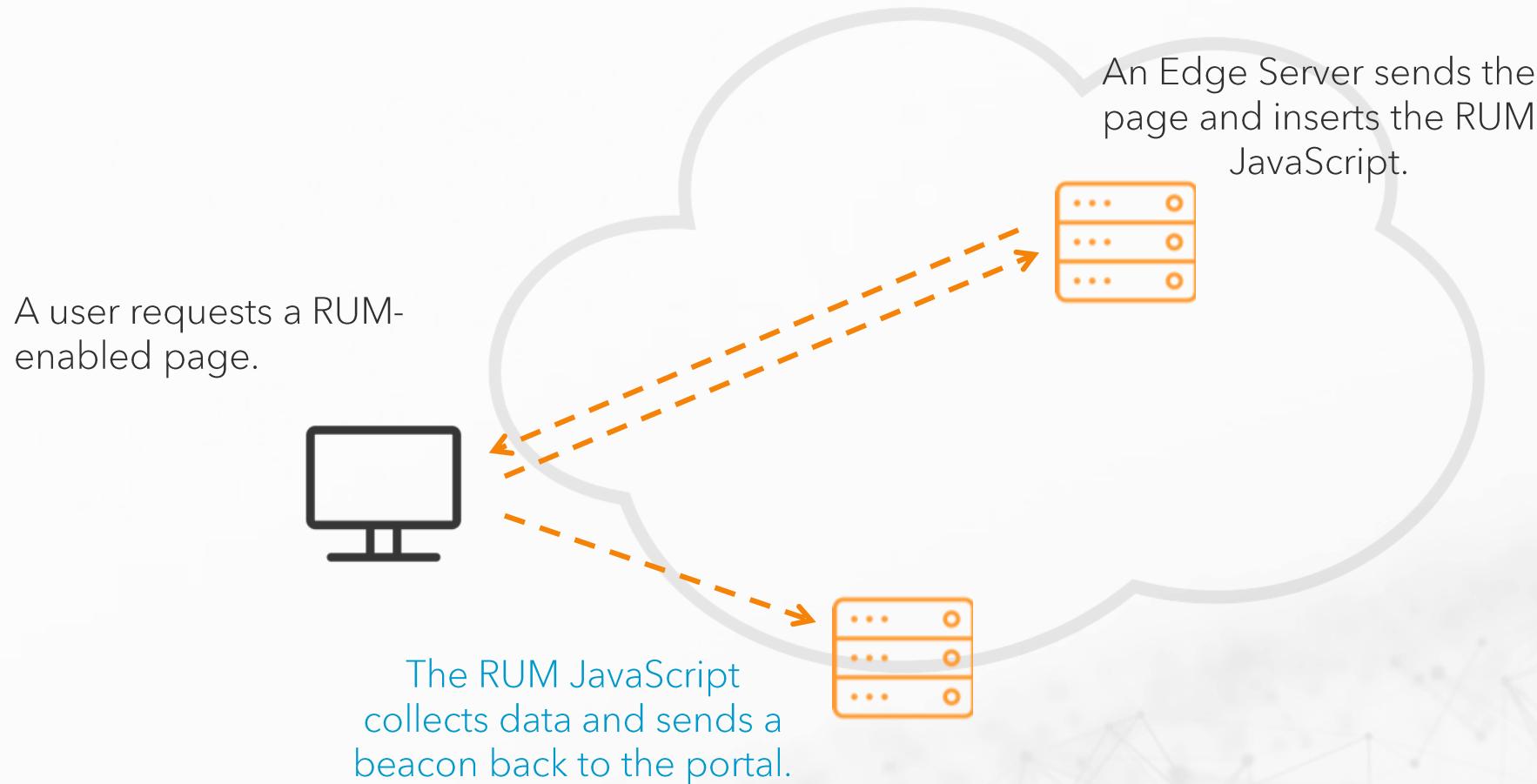
A user requests a RUM-enabled page.



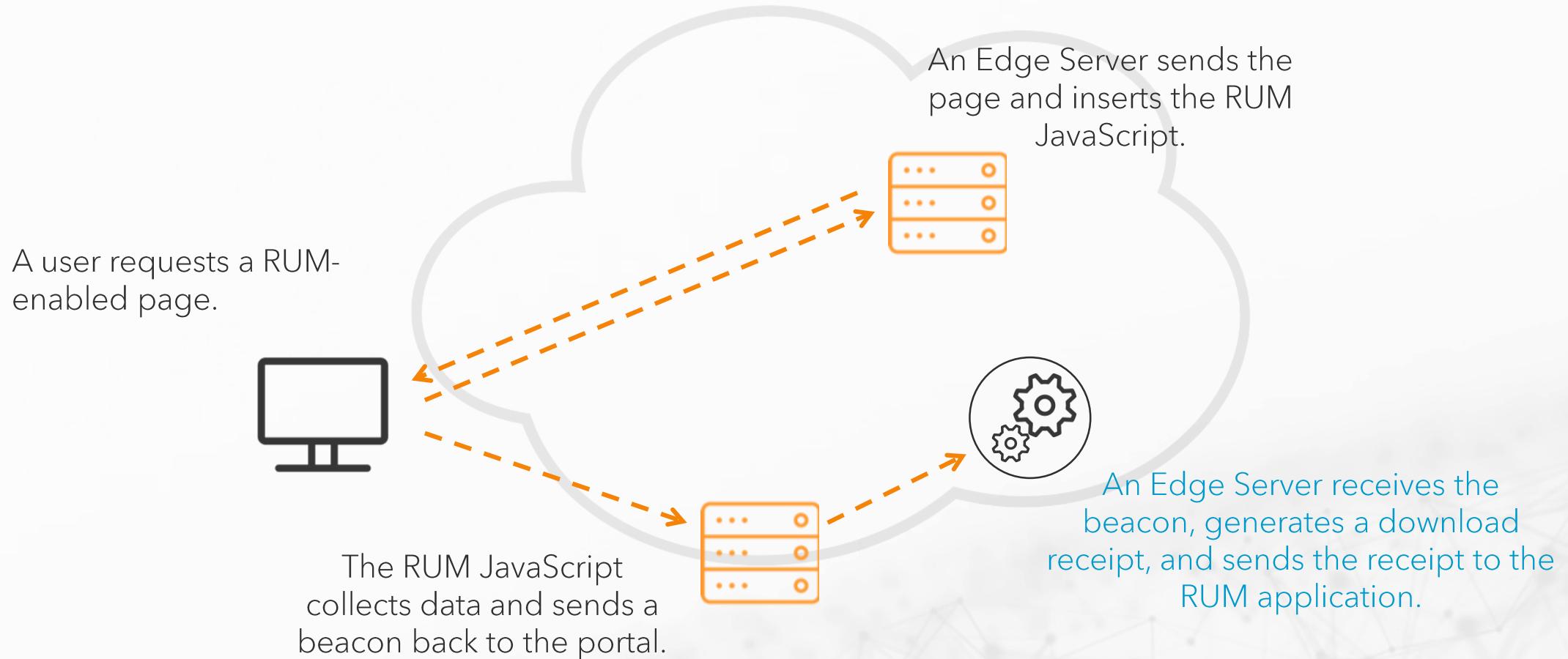
How Real User Monitoring (RUM) Works



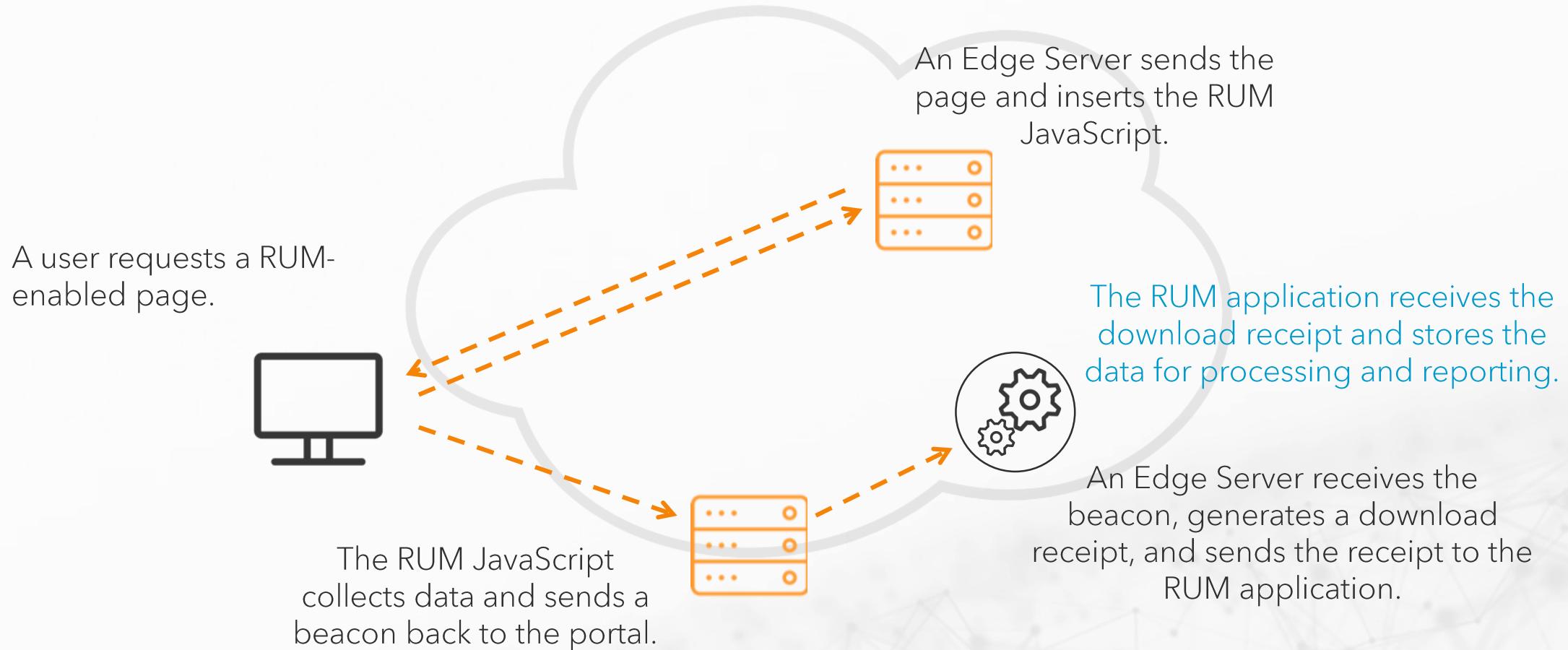
How Real User Monitoring (RUM) Works



How Real User Monitoring (RUM) Works



How Real User Monitoring (RUM) Works



Real User Monitoring Data

Metric Data

- DNS
- TCP
- First byte
- Base page download time
- First paint
- DOM content loaded
- DOM complete
- On-load event

Demographic Data

- Browser
- Operating system
- Network type and speed

Perceived Data

- Time to Visually Ready
- Time to Interactive
- Time to First Interaction
- Interactions
- Delayed Interactions
- Rage Clicks

RUM vs Synthetic Testing



When Would You Use RUM or Synthetic Measurement?

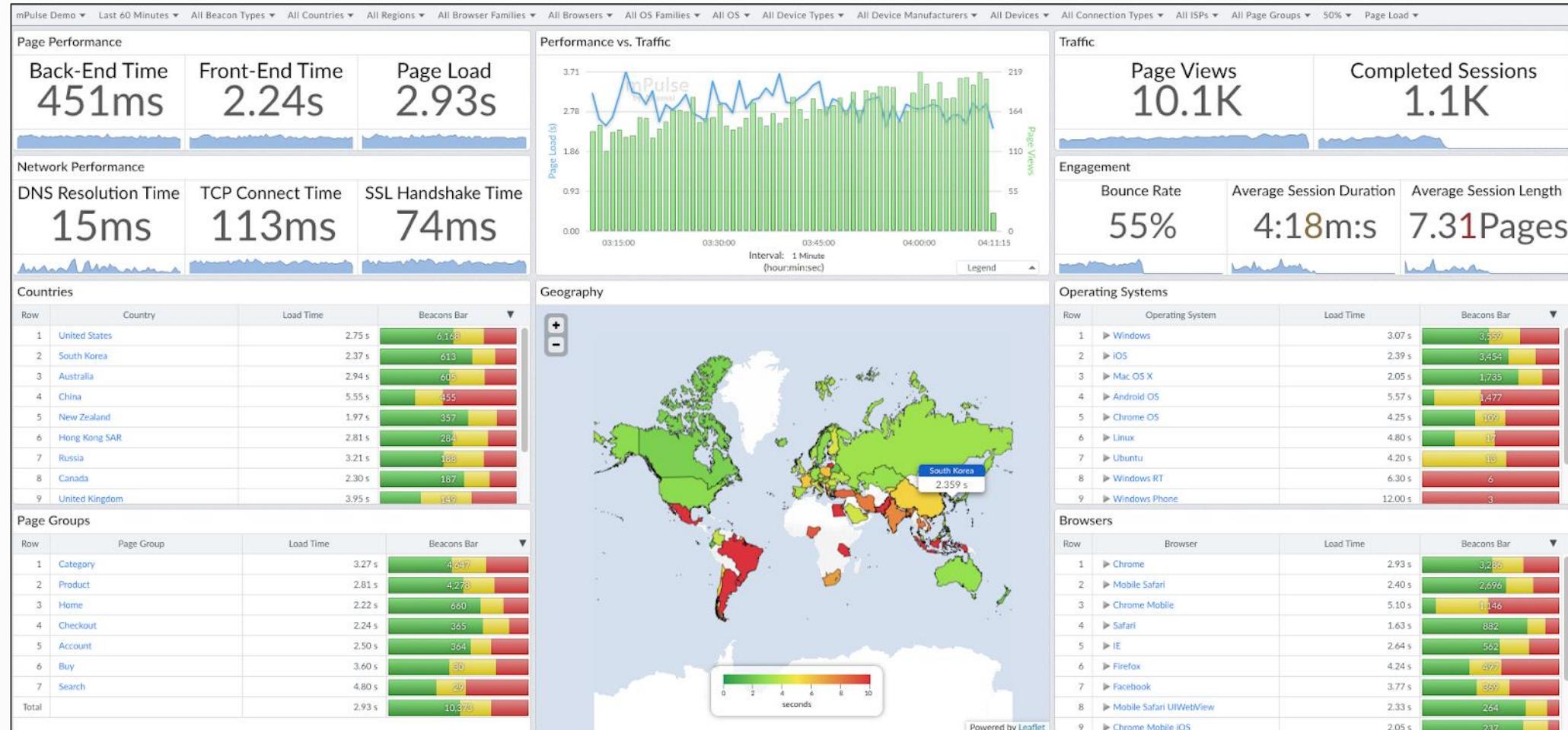
When have you used RUM and when have you used Synthetic Measurement?
Are they both part of your performance measurement activities?



mPulse - Akamai's Universal, Intelligent RUM

- Real Data, Real Time, All the Time
- Correlation and analysis
- Full SPA support
- Patented "What-If" feature
- Most matter pages identification
- Third-party resources mastering

mPulse - Akamai's Universal, Intelligent RUM



mPulse JavaScript Loader

mPulse JavaScript loader can be dynamically injected at Akamai edge or on the origin site.

Akamai Edge-Injected mPulse

Origin code Injected mPulse



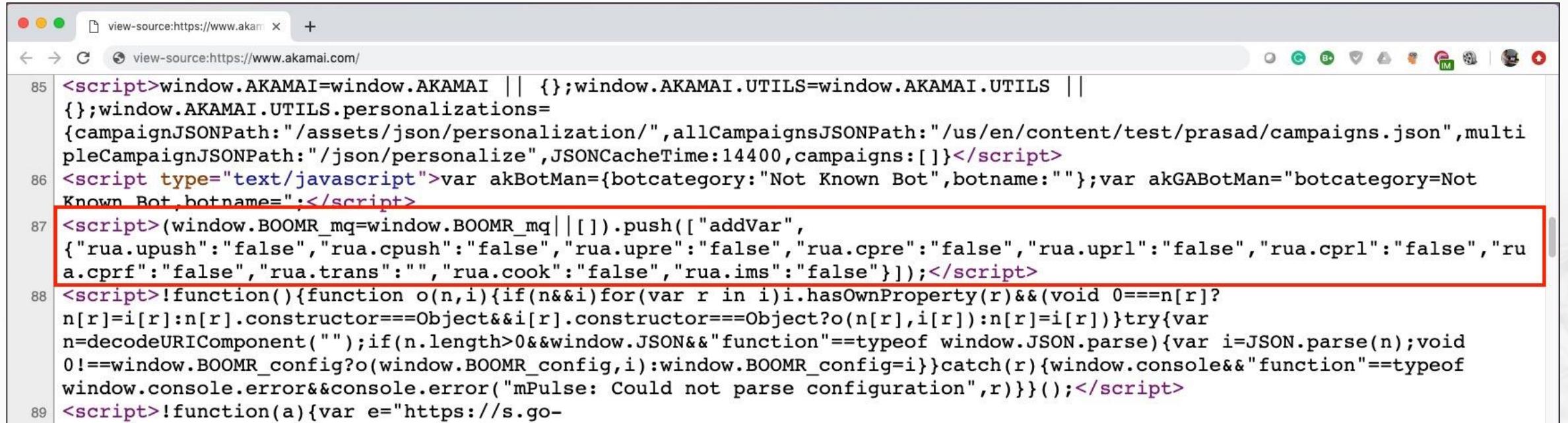
Parameters collected
via the boomerang
beacon.

Some metrics are only available for Akamai Edge-Injected mPulse

How to See mPulse in Action

Look for "boomer" that is the boomerang lib inserted by mPulse

<script>(window.BOOMR_mq=window.BOOMR_mq ...



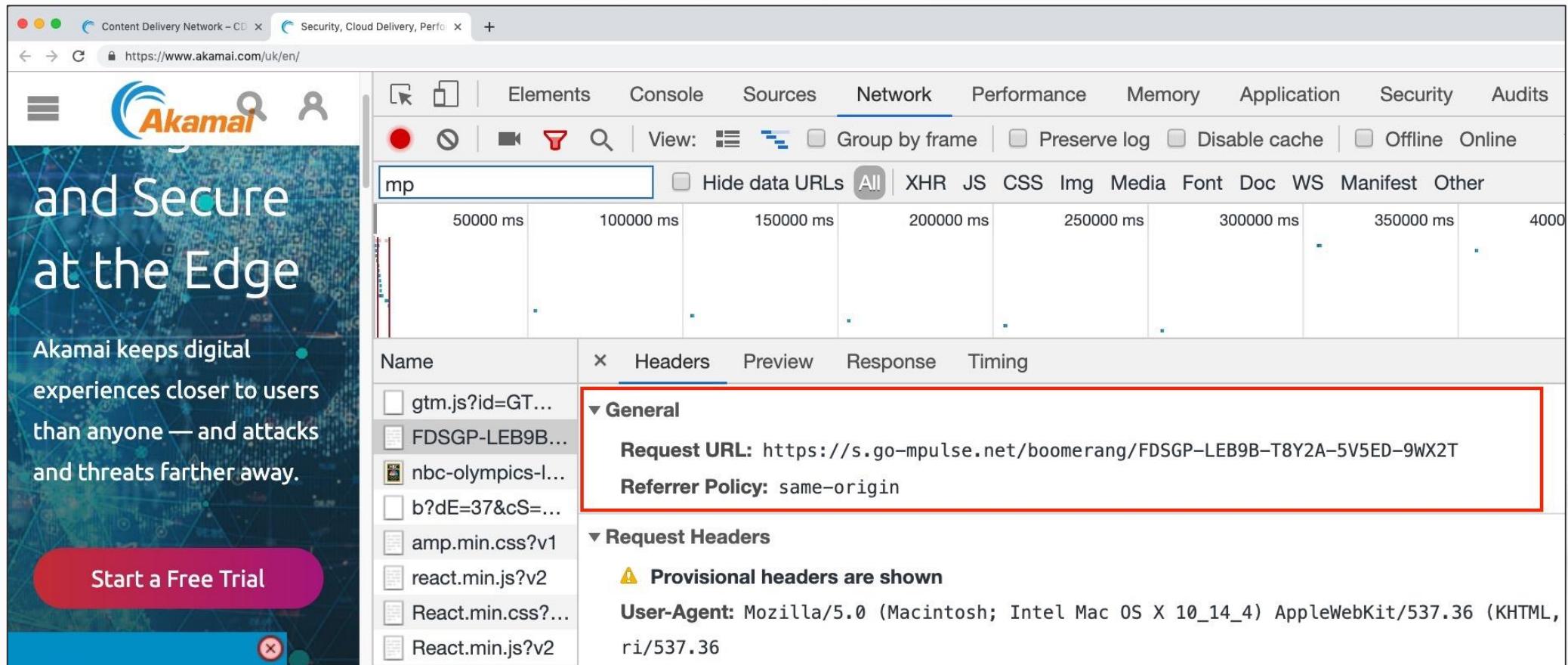
```

85 <script>window.AKAMAI=window.AKAMAI || {};window.AKAMAI.UTILS=window.AKAMAI.UTILS || {};window.AKAMAI.UTILS.personalizations={campaignJSONPath:"/assets/json/personalization/",allCampaignsJSONPath:"/us/en/content/test/prasad/campaigns.json",multipleCampaignJSONPath:"/json/personalize",JSONCacheTime:14400,campaigns:[]}</script>
86 <script type="text/javascript">var akBotMan={botcategory:"Not Known Bot",botname:""};var akGABotMan="botcategory=Not Known Bot,botname=":</script>
87 <script>(window.BOOMR_mq=window.BOOMR_mq||[]).push(["addVar",
{"rua.upush":"false","rua.cpush":"false","rua.upre":"false","rua.cpre":"false","rua.uprl":"false","rua.cprl":"false","rua.cprf":"false","rua.trans":"","rua.cook":"false","rua.ims":"false"}]);</script>
88 <script>!function(){function o(n,i){if(n&&i){for(var r in i){if(i.hasOwnProperty(r)&&(void 0==n[r])?n[r]=i[r]:n[r].constructor==Object&&i[r].constructor==Object?o(n[r],i[r]):n[r]=i[r])}}try{var n=decodeURIComponent("");if(n.length>0&&window.JSON&&"function"==typeof window.JSON.parse){var i=JSON.parse(n);void 0!=window.BOOMR_config?o(window.BOOMR_config,i):window.BOOMR_config=i}}catch(r){window.console&&"function"==typeof window.console.error&&window.console.error("mPulse: Could not parse configuration",r)}}();</script>
89 <script>!function(a){var e="https://s.go-

```

How to See mPulse in Action

You will then see a call for the mPulse beacon.



The screenshot shows a browser window with the URL <https://www.akamai.com/uk/en/>. The page content features the Akamai logo and the text "Secure and Secure at the Edge". Below this, a paragraph reads: "Akamai keeps digital experiences closer to users than anyone — and attacks and threats farther away." A red button labeled "Start a Free Trial" is visible at the bottom. The browser's developer tools Network tab is open, showing a list of network requests. One request, "FDSGP-LEB9B...", is highlighted with a red box. The "Headers" tab is selected for this request. The "General" section shows the "Request URL: <https://s.go-mpulse.net/boomerang/FDSGP-LEB9B-T8Y2A-5V5ED-9WX2T>" and "Referrer Policy: same-origin". The "Request Headers" section includes a warning: "⚠ Provisional headers are shown" and lists the "User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36".

Test Your Akamai Edge!

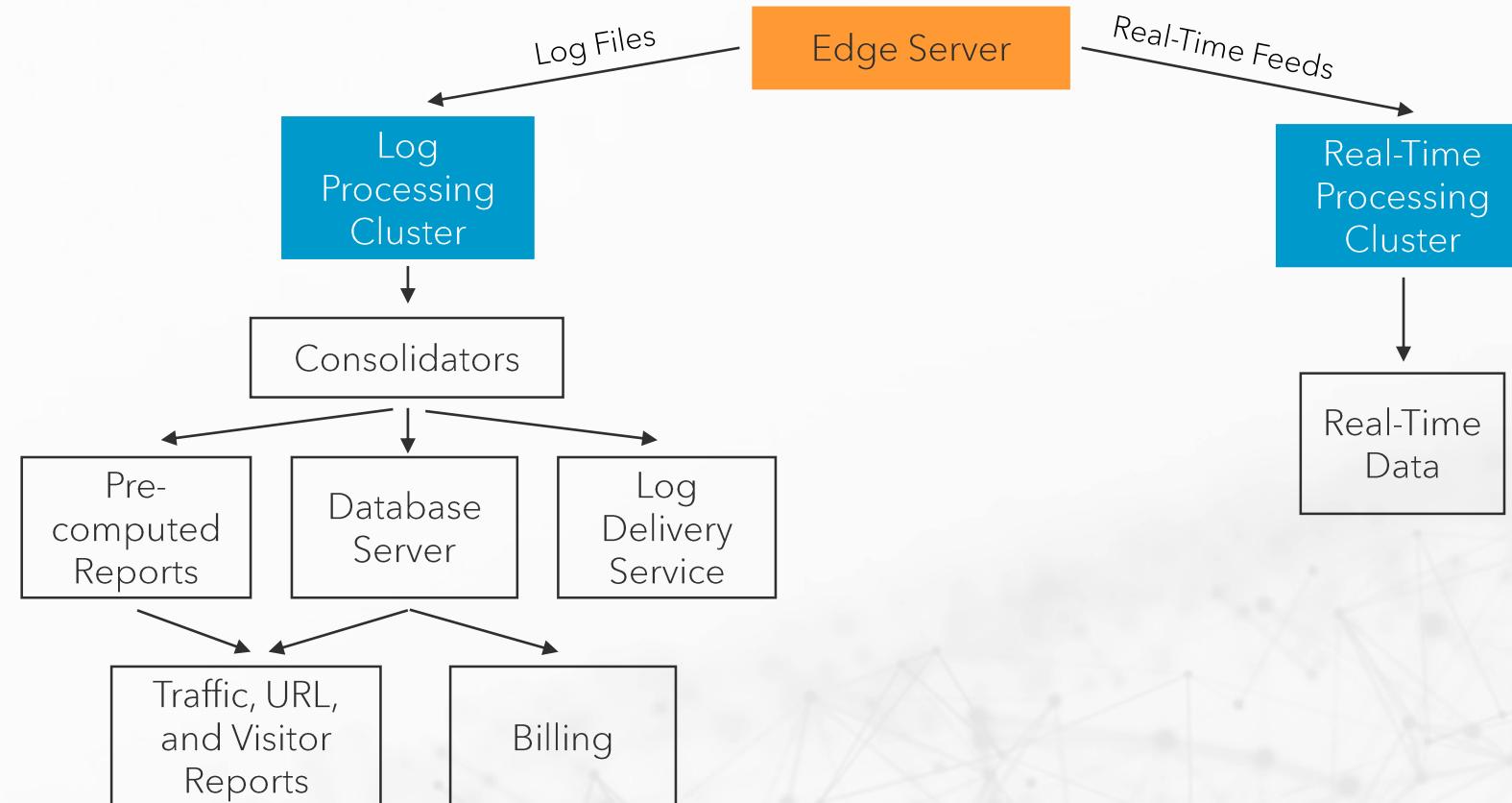
How does mPulse provide an accurate picture of end-user experience?

- A. It uses a Javascript beacon to collect timing data from the end-user's browser and sends it to Akamai for analysis and reporting.
- B. It collects performance data from multiple networks distributed globally and sends it back to Akamai analytics platform.
- C. It sends data to Akamai or the customer's data center for analysis and reporting
- D. It gathers, analyzes, and reports data in the Akamai Control Center.

Akamai Control Center - Reports

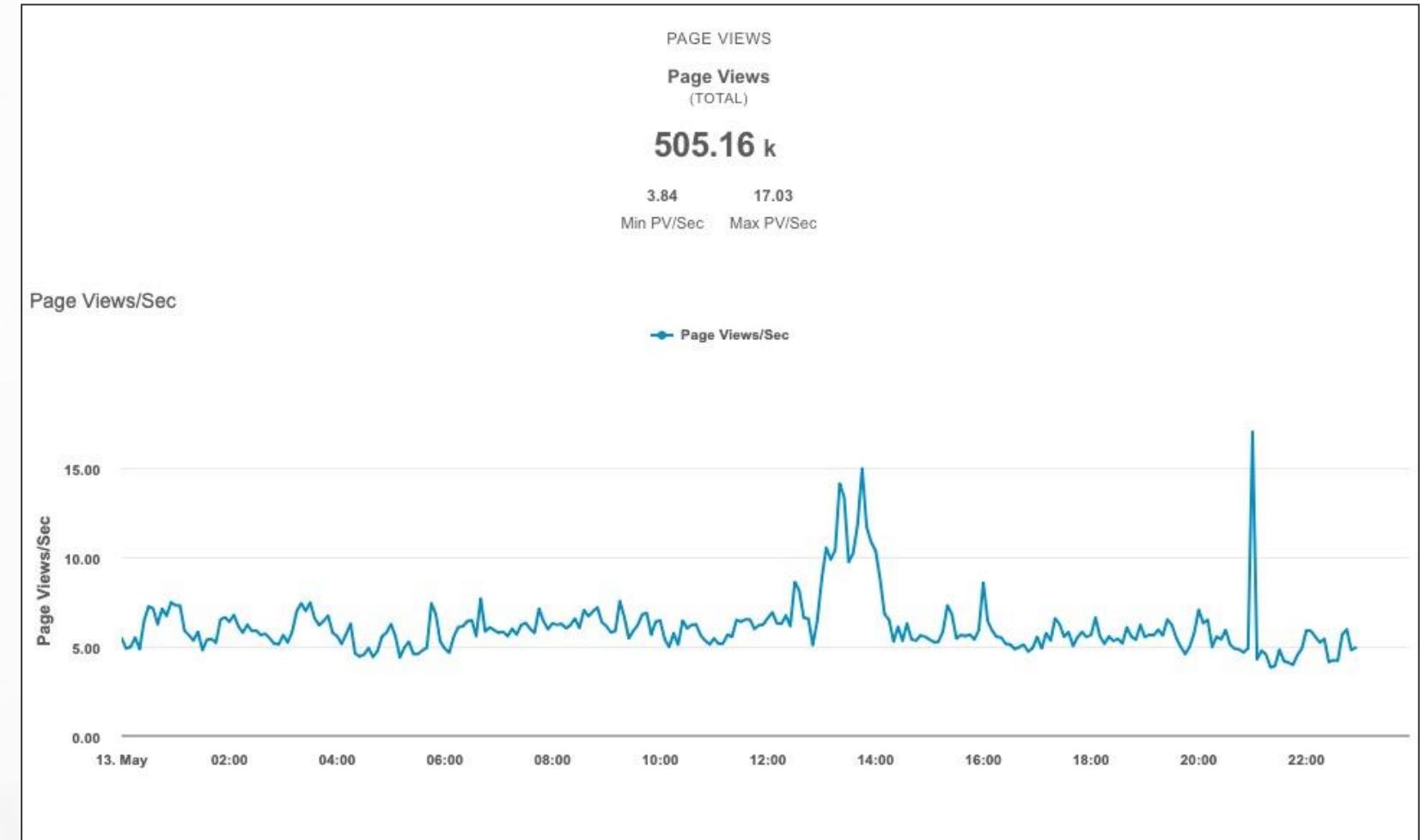
Where Does Data Come From?

Akamai servers periodically send real-time data and web server logs to Akamai's Distributed Data Collection system (DDC).



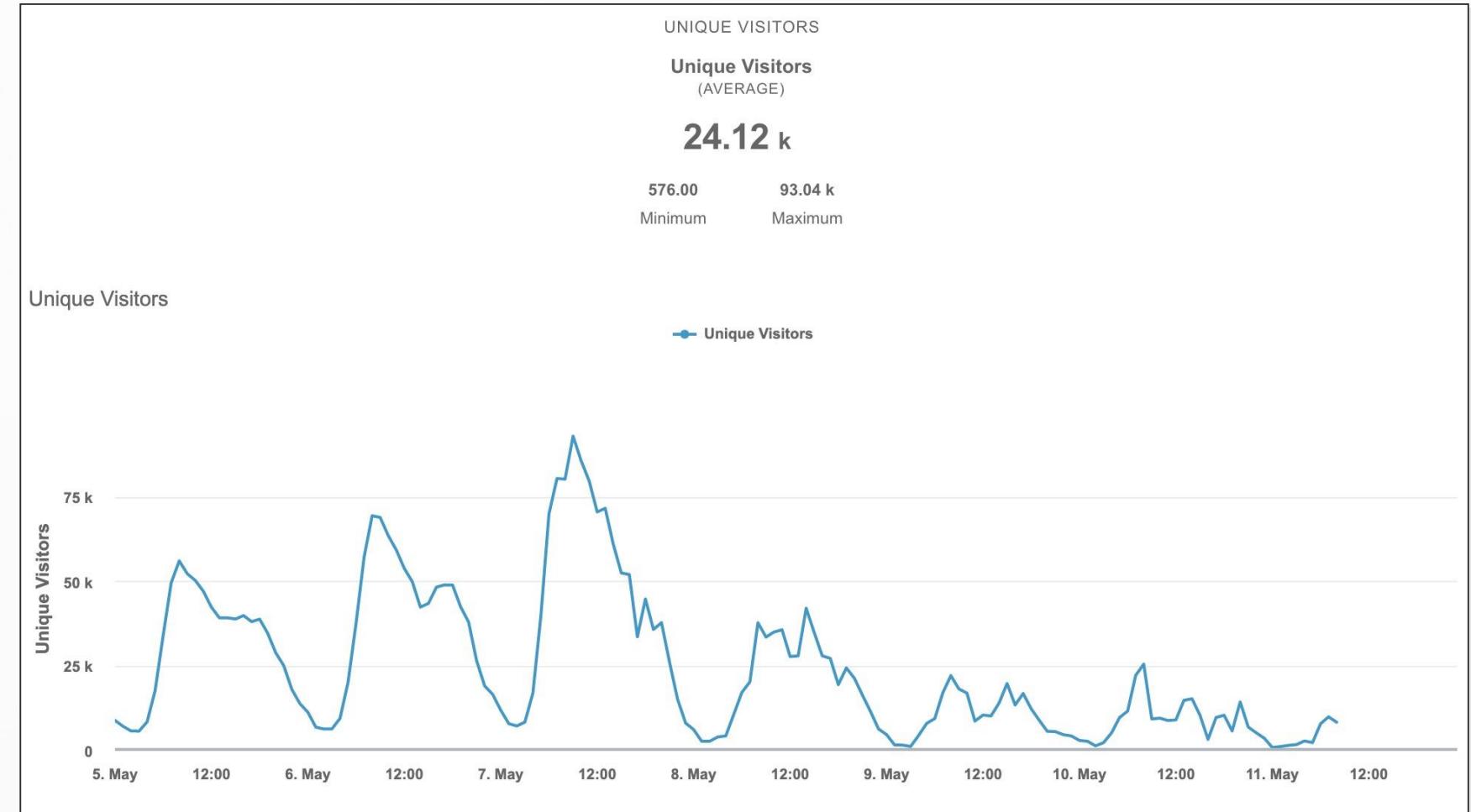
User Traffic Report

The User Traffic Report gives an easy overview of the standard traffic metrics: page views, edge hits, edge bandwidth, and how users break down by browser use, operating system use, and geographically.



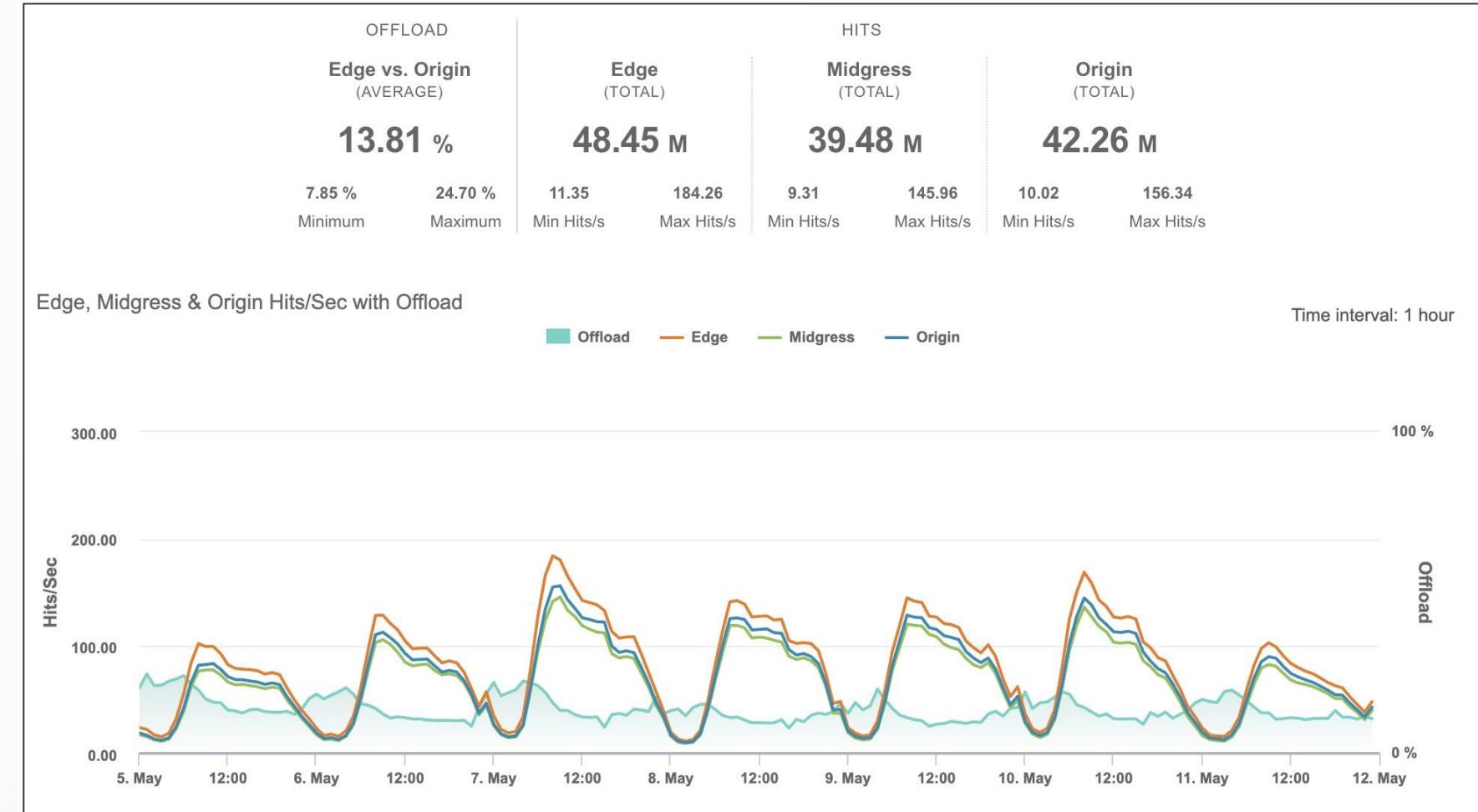
Unique Visitors Report

The Unique Visitors report gives you insight into the users generating your traffic, tracking unique IP/user agent combinations over time, geographically, and by browser or operating system.



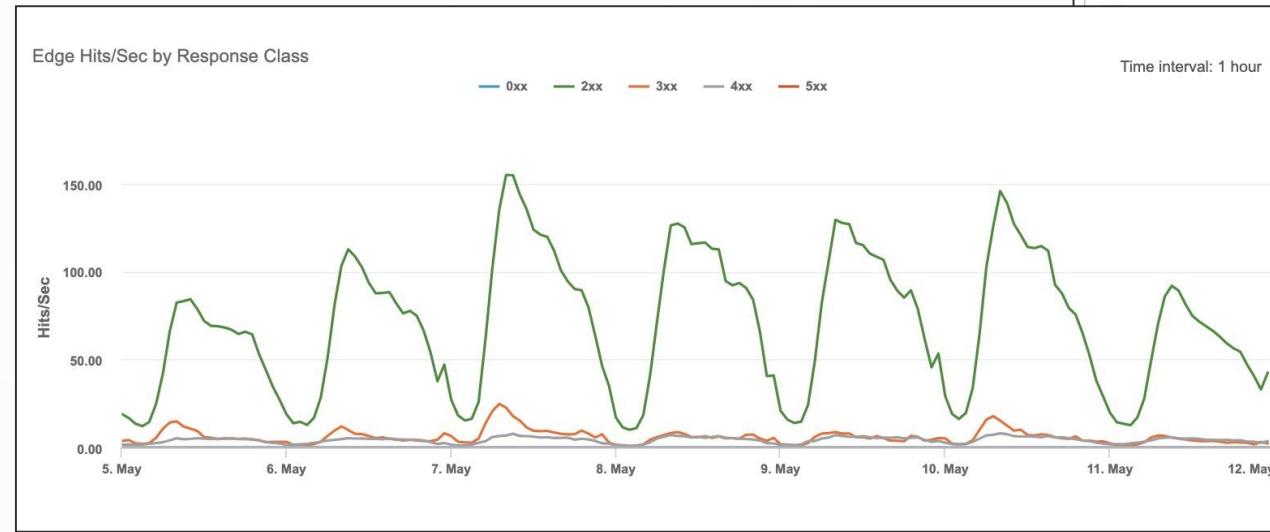
Traffic Report

The Traffic report provides an aggregate view of your traffic and offload, how much Akamai benefits your content delivery. You can see edge and origin hits and volume, as well as the percentage of traffic directed away from your origin and onto the Akamai platform.



Responses Report

The Responses report gives you insight into the response codes your content is returning.

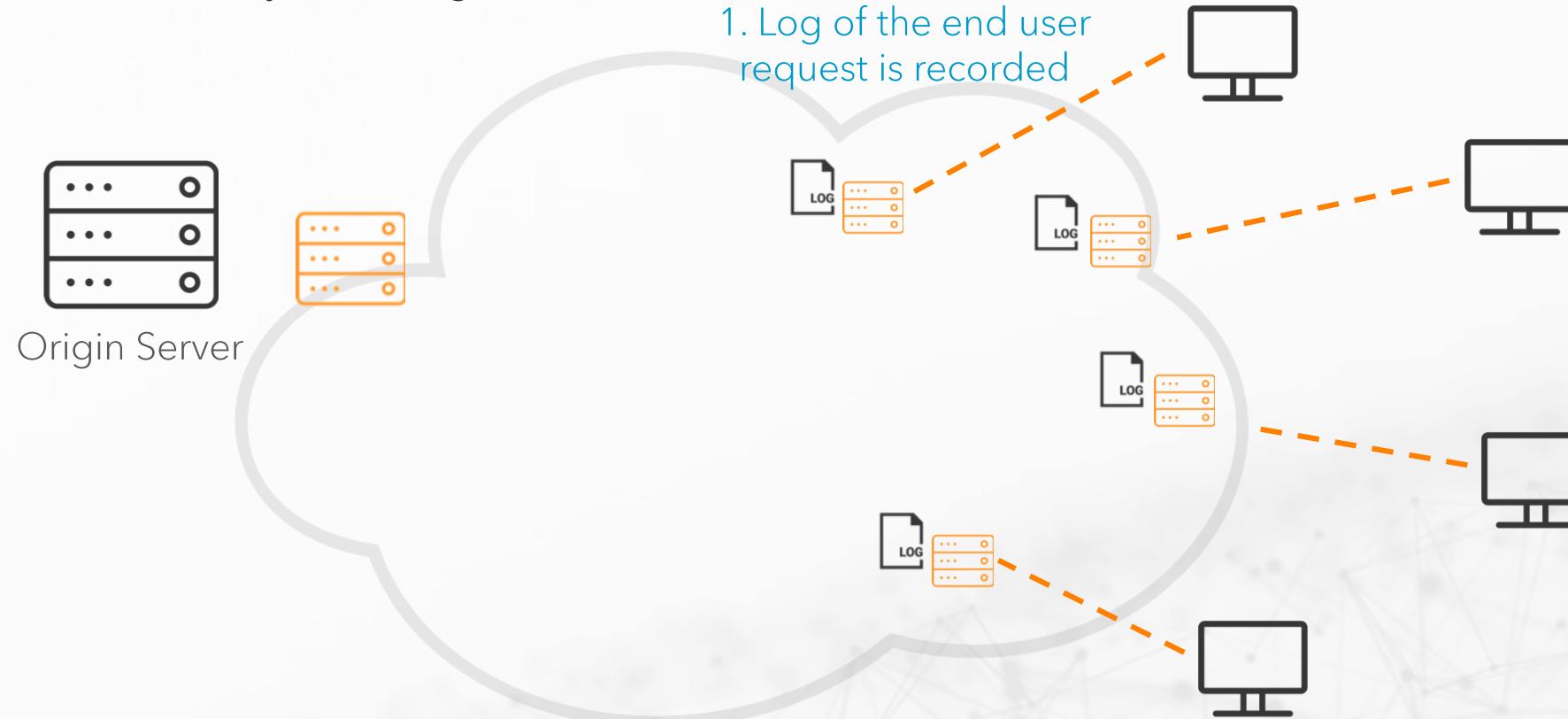


Edge & Origin Hits by Response

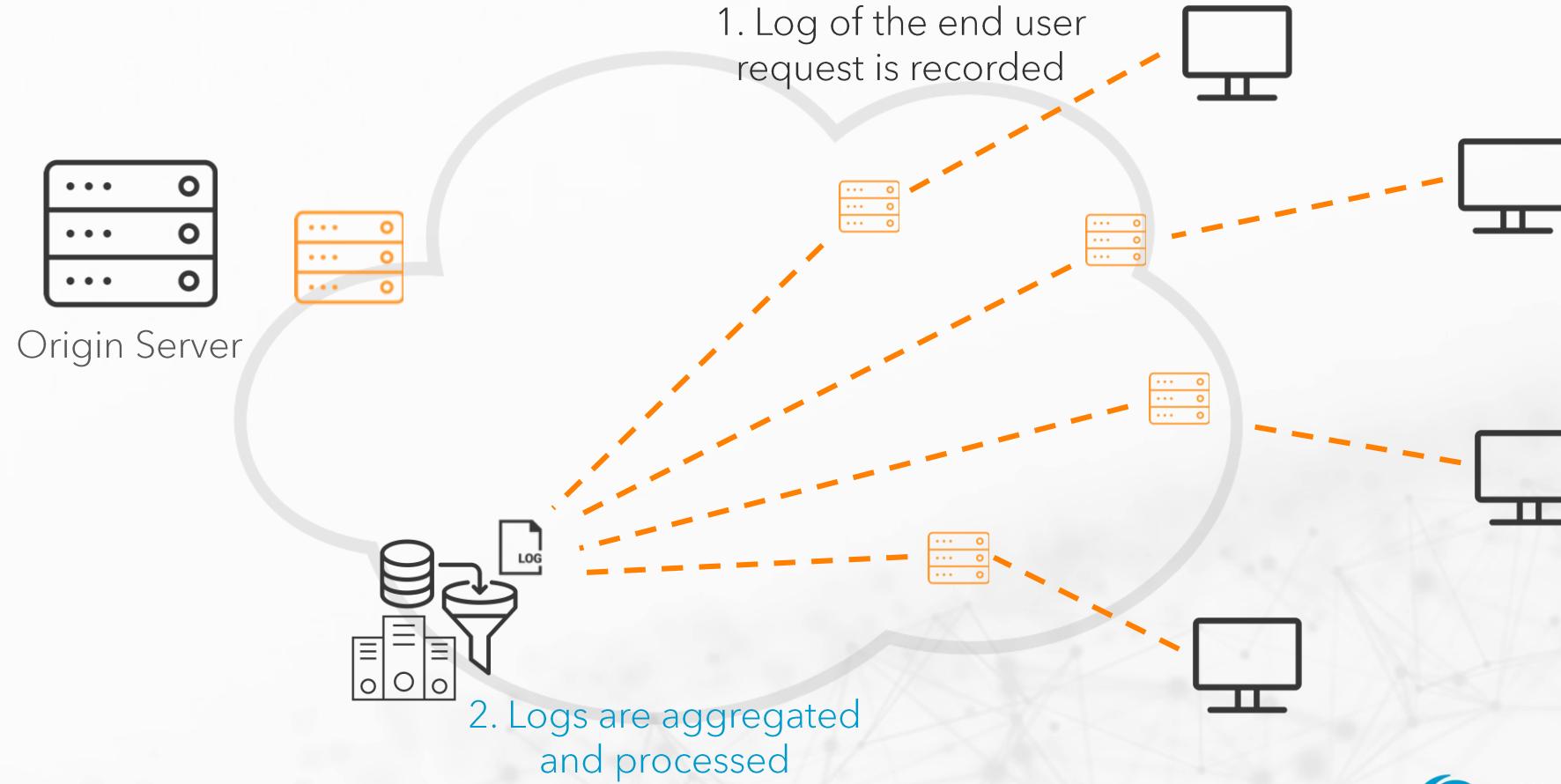
| Response Code | Edge Hits | Edge Hits % | Origin Hits | Origin Hits % |
|--------------------------|------------|-------------|-------------|---------------|
| 200 | 42,197,156 | 87.34 % | 41,877,889 | 99.13 % |
| 301 | 3,526,029 | 7.30 % | 62,319 | 0.15 % |
| 403 | 2,342,537 | 4.85 % | 0 | 0.00 % |
| 404 | 173,020 | 0.36 % | 173,742 | 0.41 % |
| 302 | 30,958 | 0.06 % | 31,080 | 0.07 % |
| 304 | 20,324 | 0.04 % | 20,427 | 0.05 % |
| 206 | 19,552 | 0.04 % | 19,460 | 0.05 % |
| | 2,154 | 0.00 % | 0 | 0.00 % |
| | 555 | 0.00 % | 559 | 0.00 % |
| | 140 | 0.00 % | 0 | 0.00 % |
| | 78 | 0.00 % | 61,858 | 0.15 % |
| | 50 | 0.00 % | 50 | 0.00 % |
| | 46 | 0.00 % | 41 | 0.00 % |
| | 5 | 0.00 % | 5 | 0.00 % |
| | 4 | 0.00 % | 0 | 0.00 % |
| | 1 | 0.00 % | 1 | 0.00 % |
| Items Per Page: 10 25 50 | | | | |

Log Delivery Service

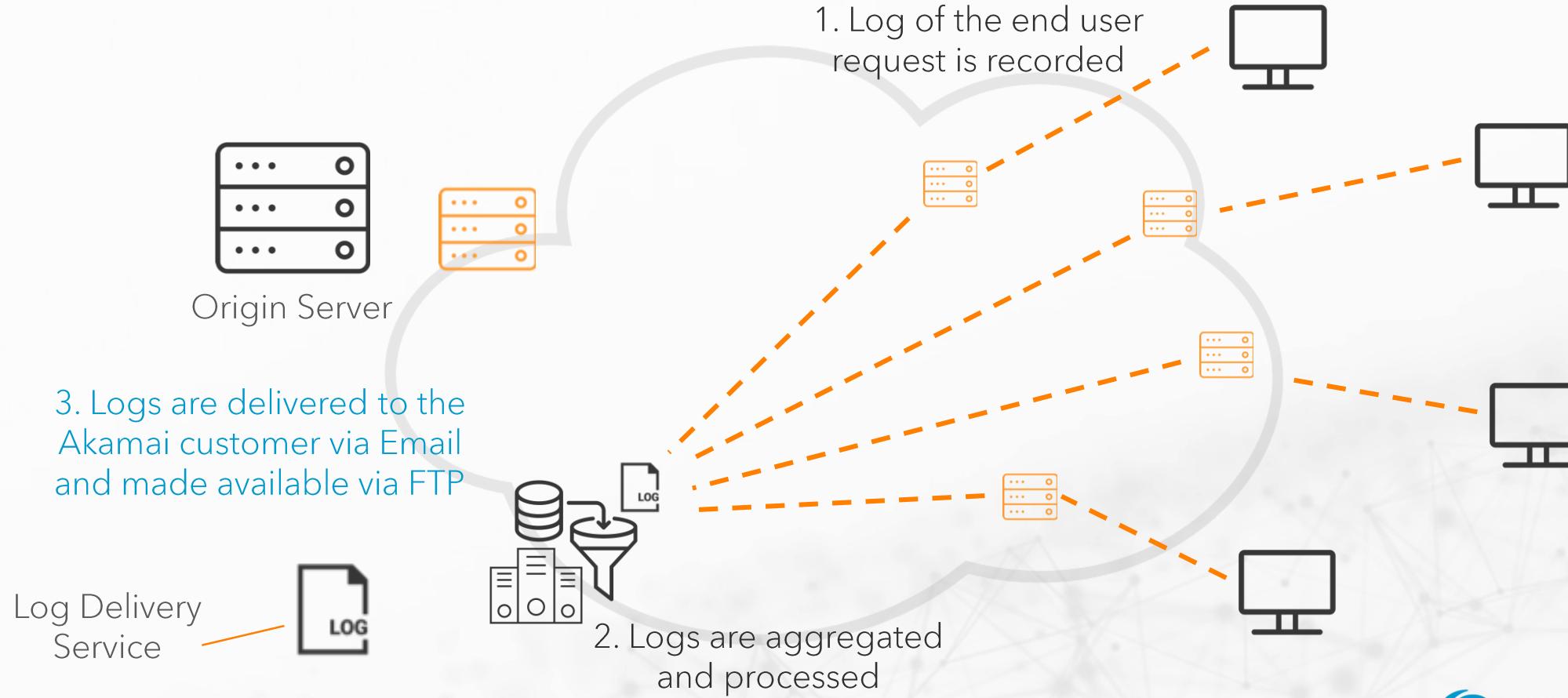
Akamai's Log Delivery Service provides customers with Akamai Edge server logs from the various Akamai services that they are using.



Log Delivery Service



Log Delivery Service



What Is Cloud Monitor?

Cloud Monitor extends Log Deliver Service with a real-time, push API that delivers the same critical transaction and security event data to the customer. Cloud Monitor endpoints include Splunk and Sumo Logic.

The Cloud Monitor has four key components:

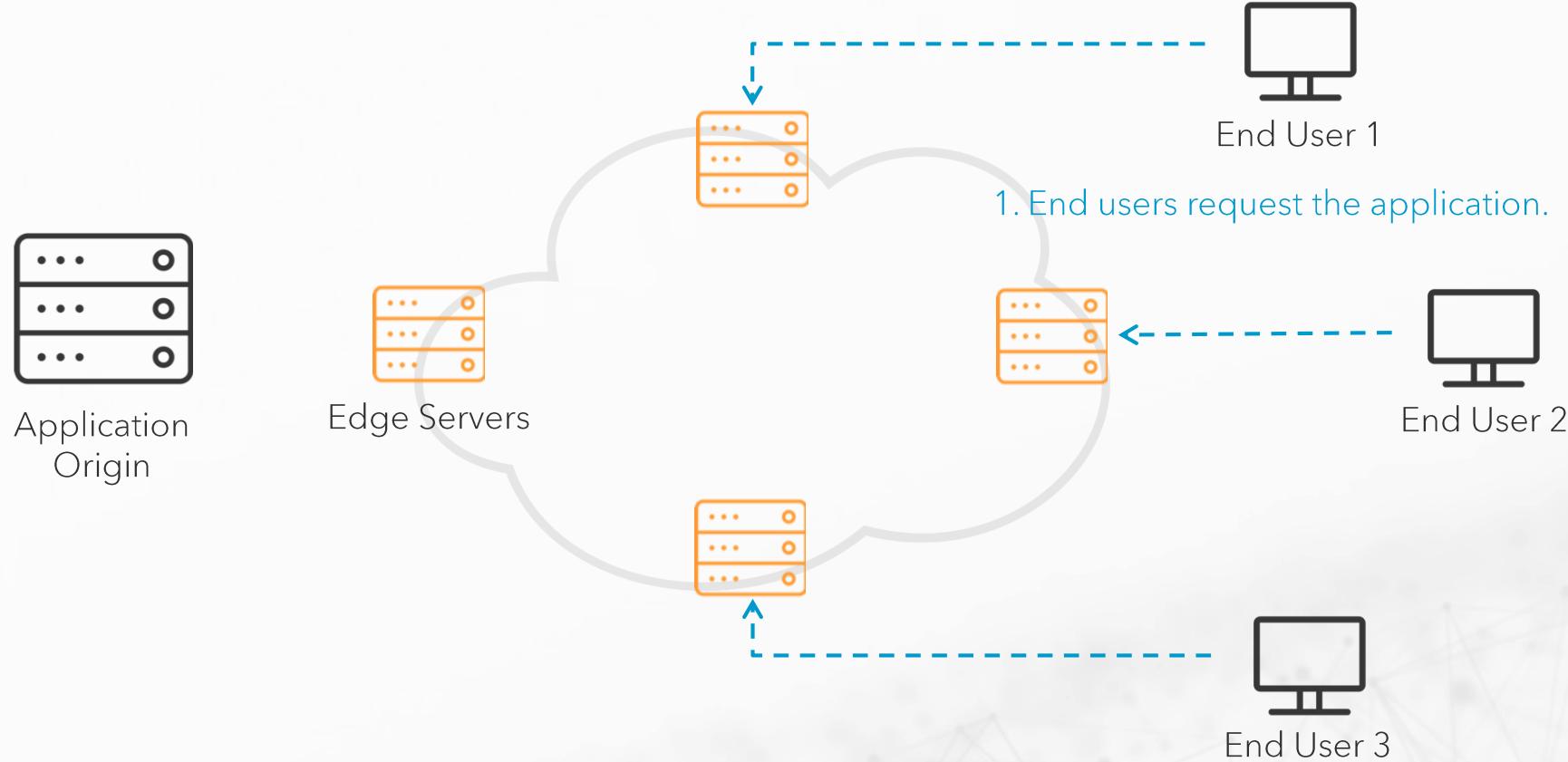
Data
Delivery
Property

Akamaized
Property

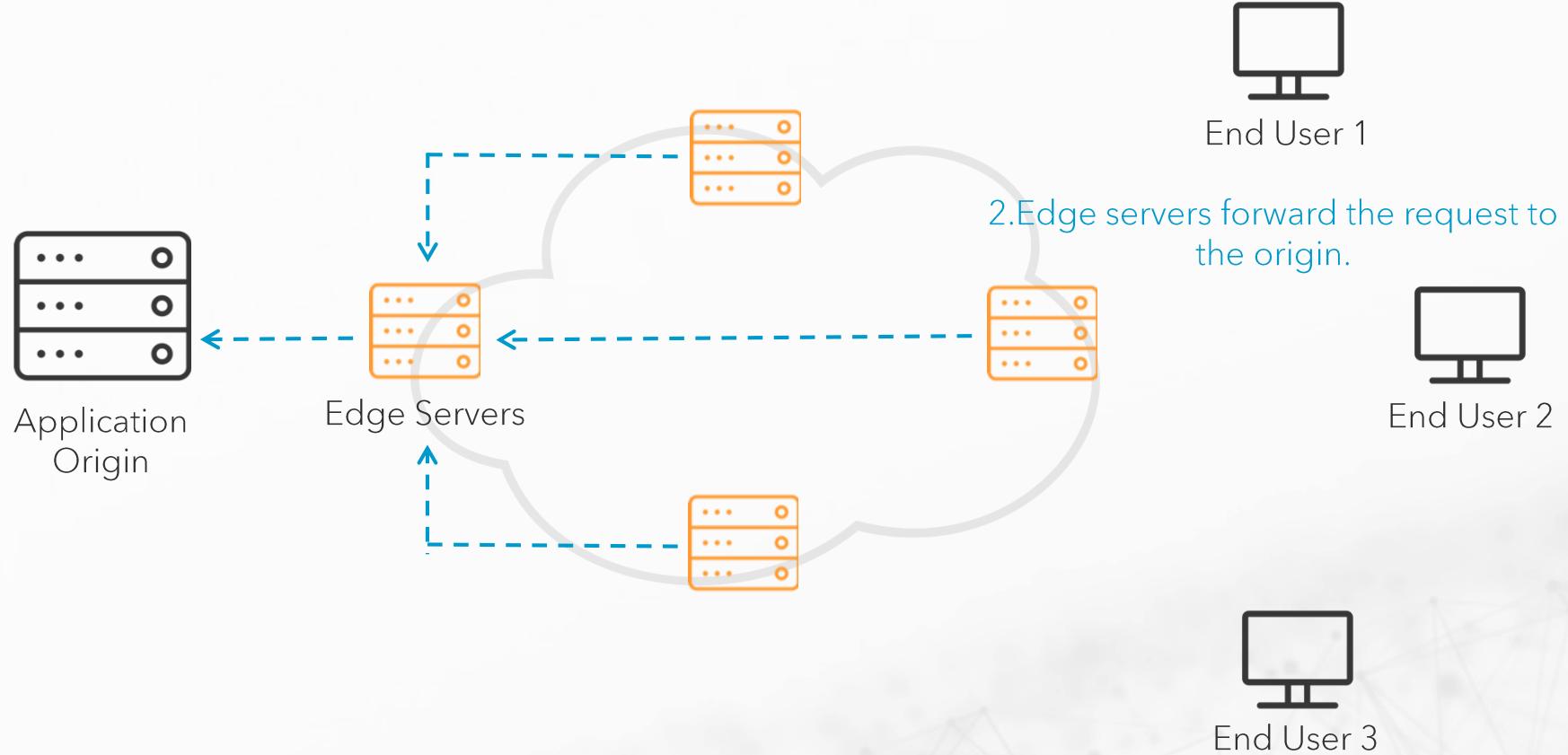
Application
Origin

Log Delivery
EndPoint

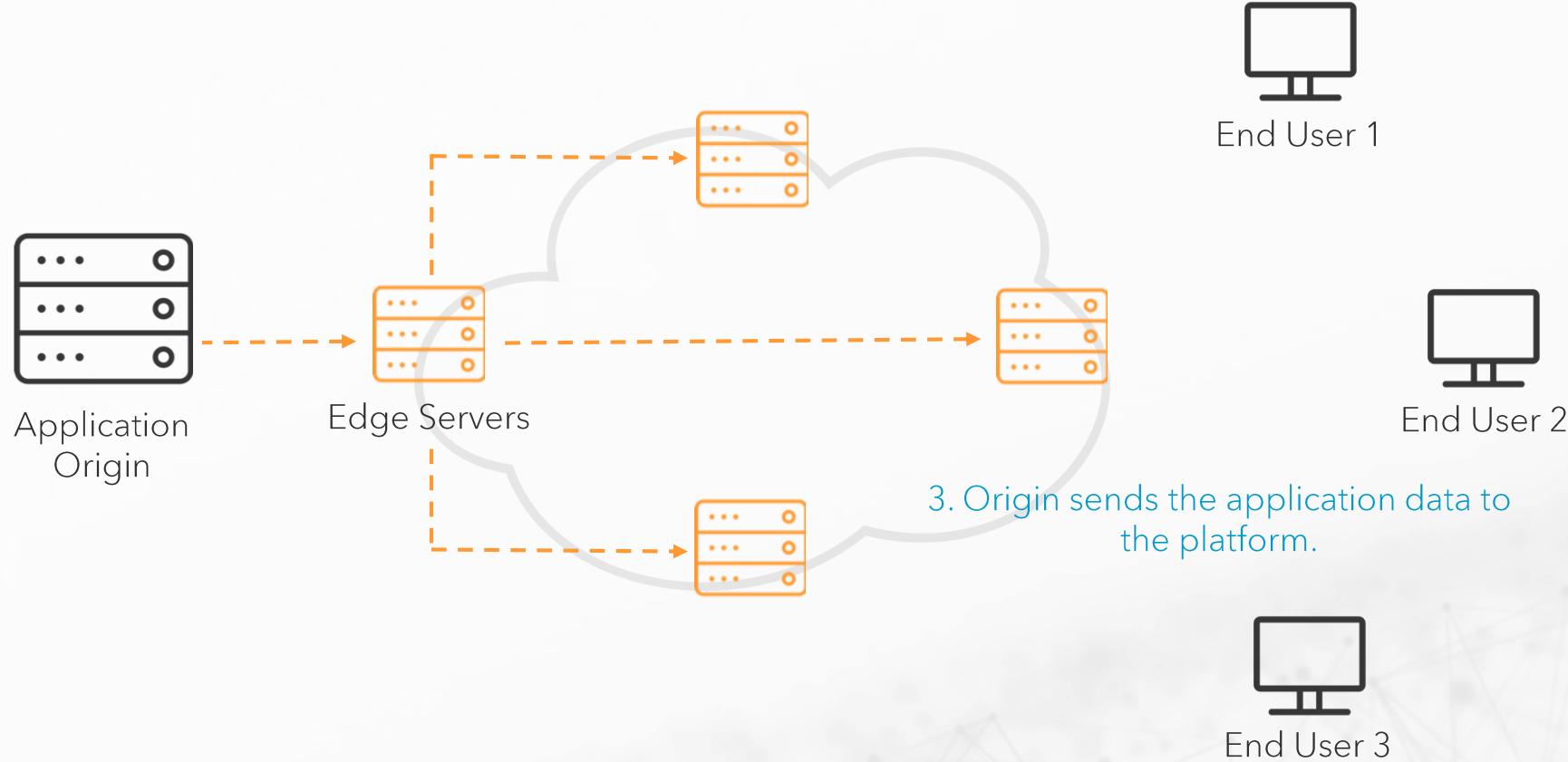
Cloud Monitor - How It Works



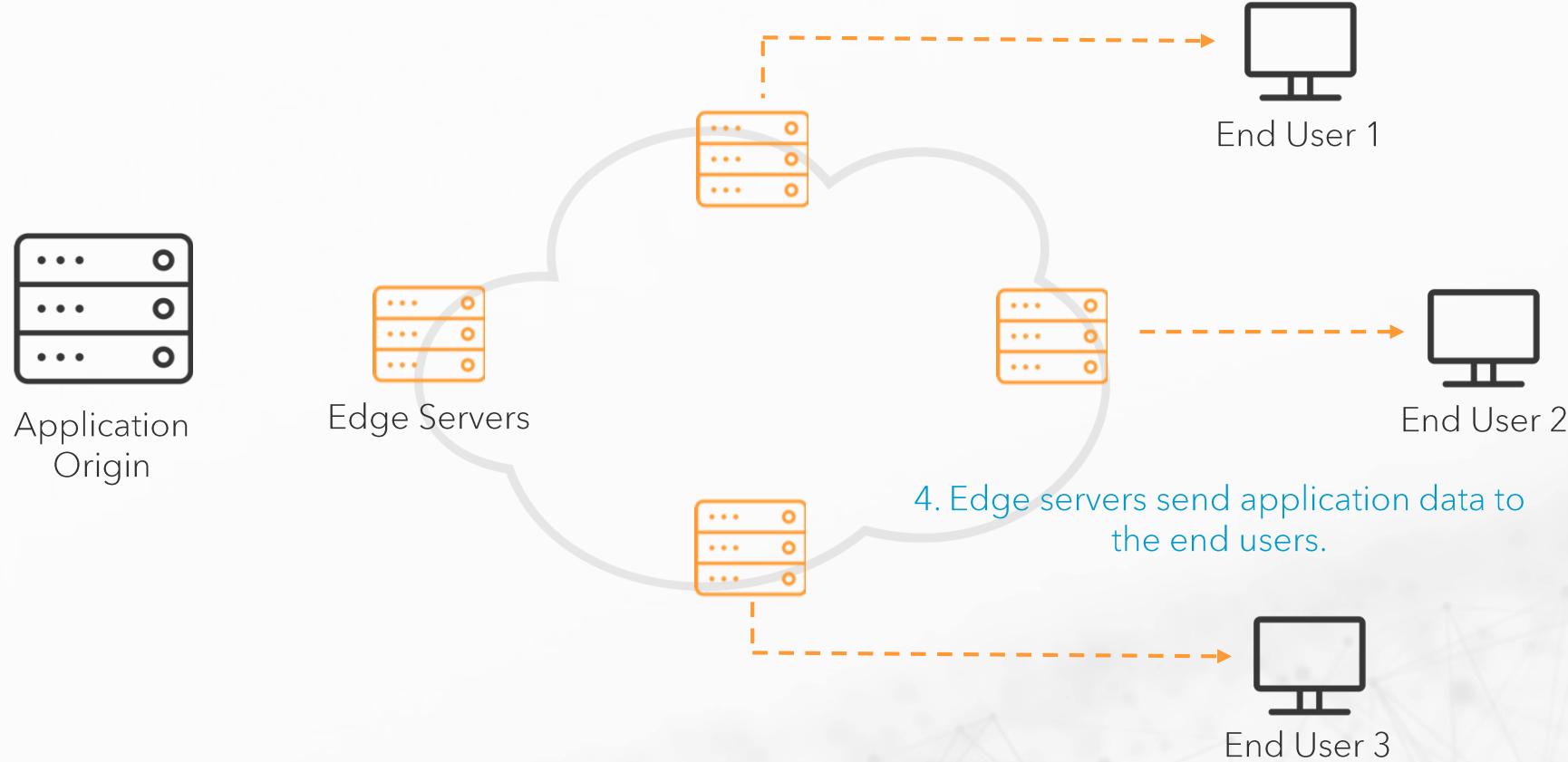
Cloud Monitor - How It Works



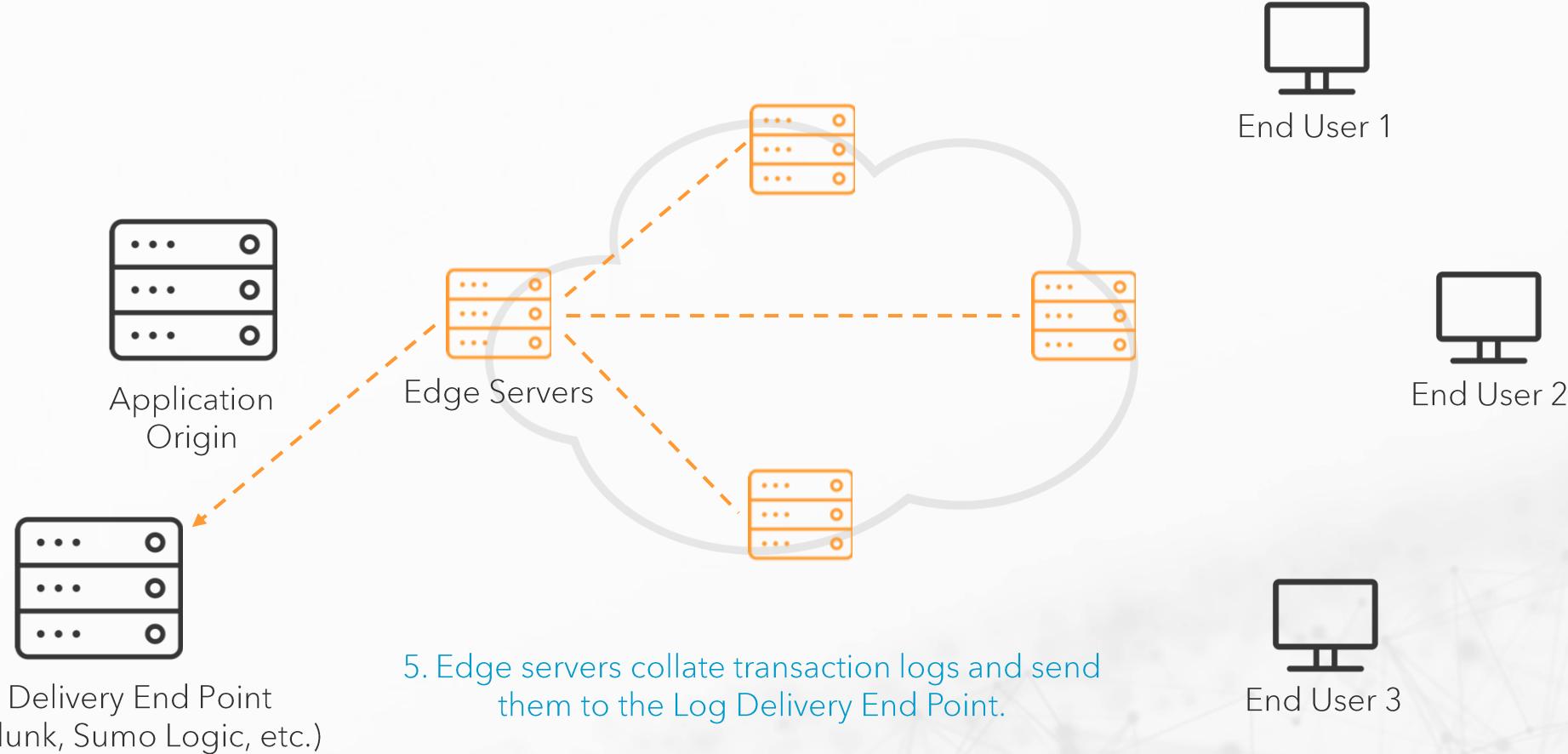
Cloud Monitor - How It Works



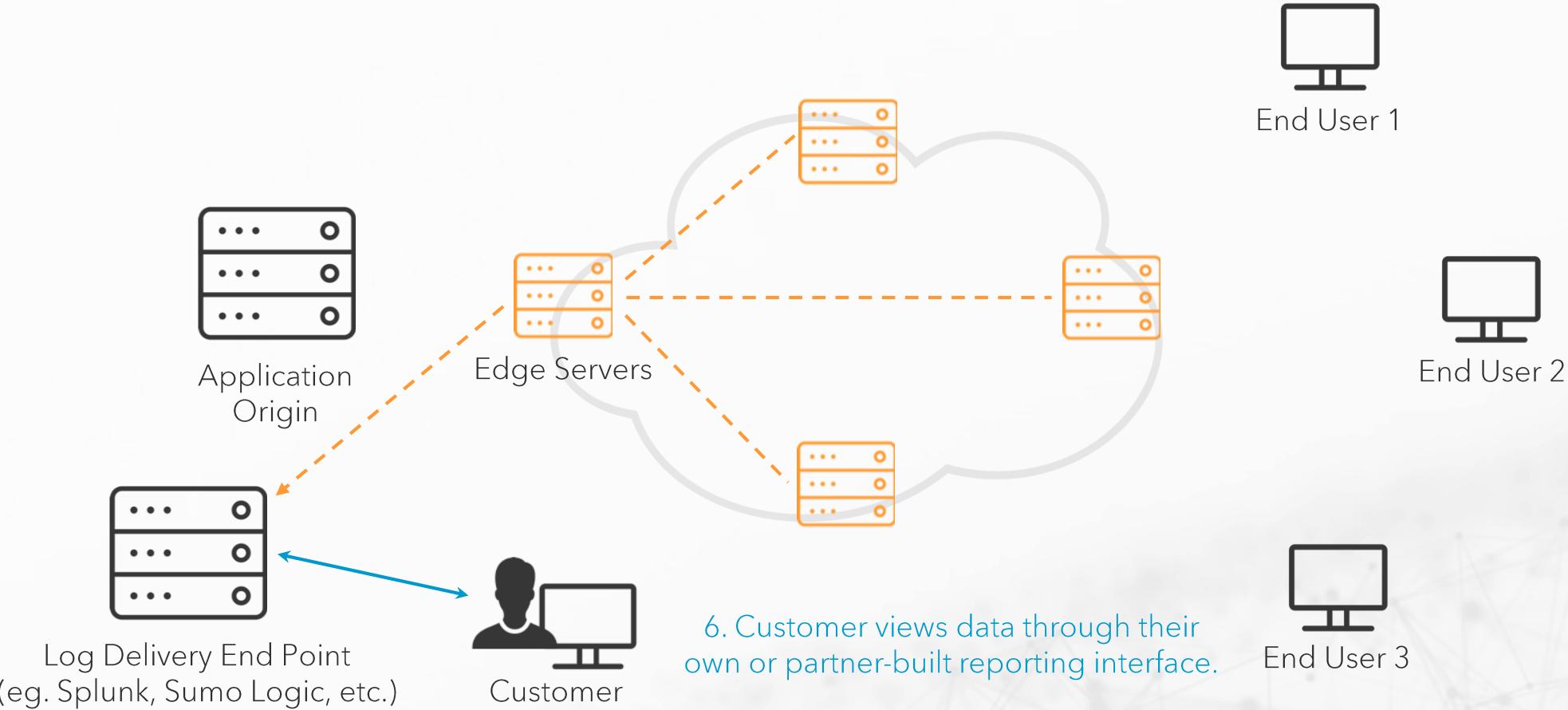
Cloud Monitor - How It Works



Cloud Monitor - How It Works



Cloud Monitor - How It Works



Test Your Akamai Edge!

How does mPulse provide an accurate picture of end-user experience?

- A. It gives you offload data indicating how much traffic is being served by Akamai without going to origin.
- B. It provides customers with Akamai Edge server logs representing the individual requests/responses from end users to the Edge.
- C. It gives the customer details on bytes saved from Adaptive Image Compression.
- D. It provides customers with user traffic reports, which gives them with an overview of the standard traffic metrics.

DEMO

Log Delivery Service



Q&A



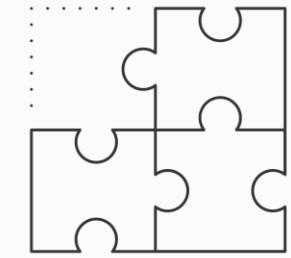
Summary

Reports

- Aggregated data and server logs captured by the Edge servers on the Akamai Intelligent Platform.
- These reports are viewable on Akamai Control Center.

Log Delivery Service

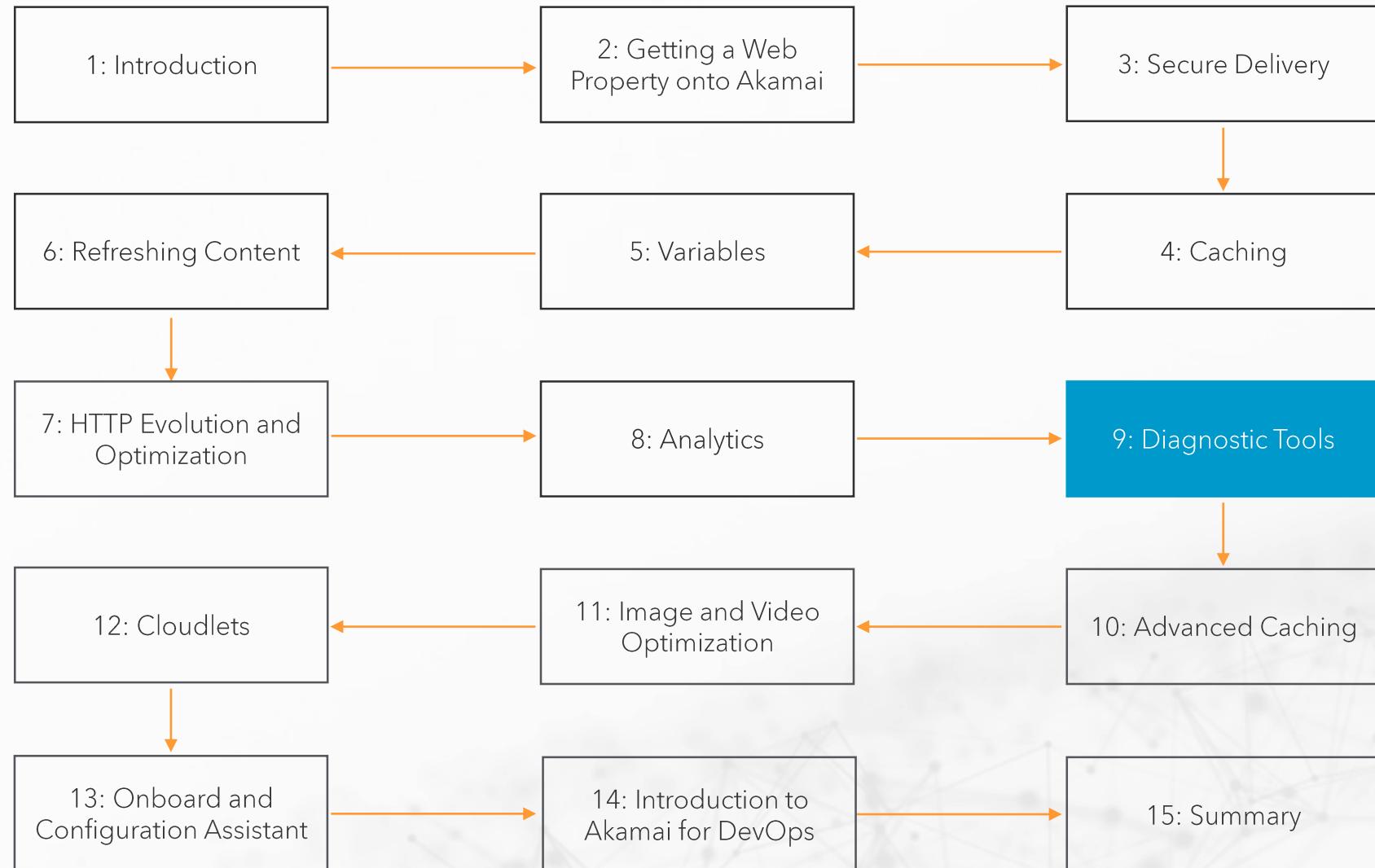
A service that provides customers with Akamai Edge server logs from the various Akamai services that they are using



Real User Monitoring

A tool that provides you with data about the performance of your websites, based on real end-user experiences

Agenda



MODULE 9

Diagnostic Tools



Module Objectives

At the end of this module, you will be able to:

- List diagnostic tools built into Akamai Control Center.
- Describe Akamai specific tools.



DEMO

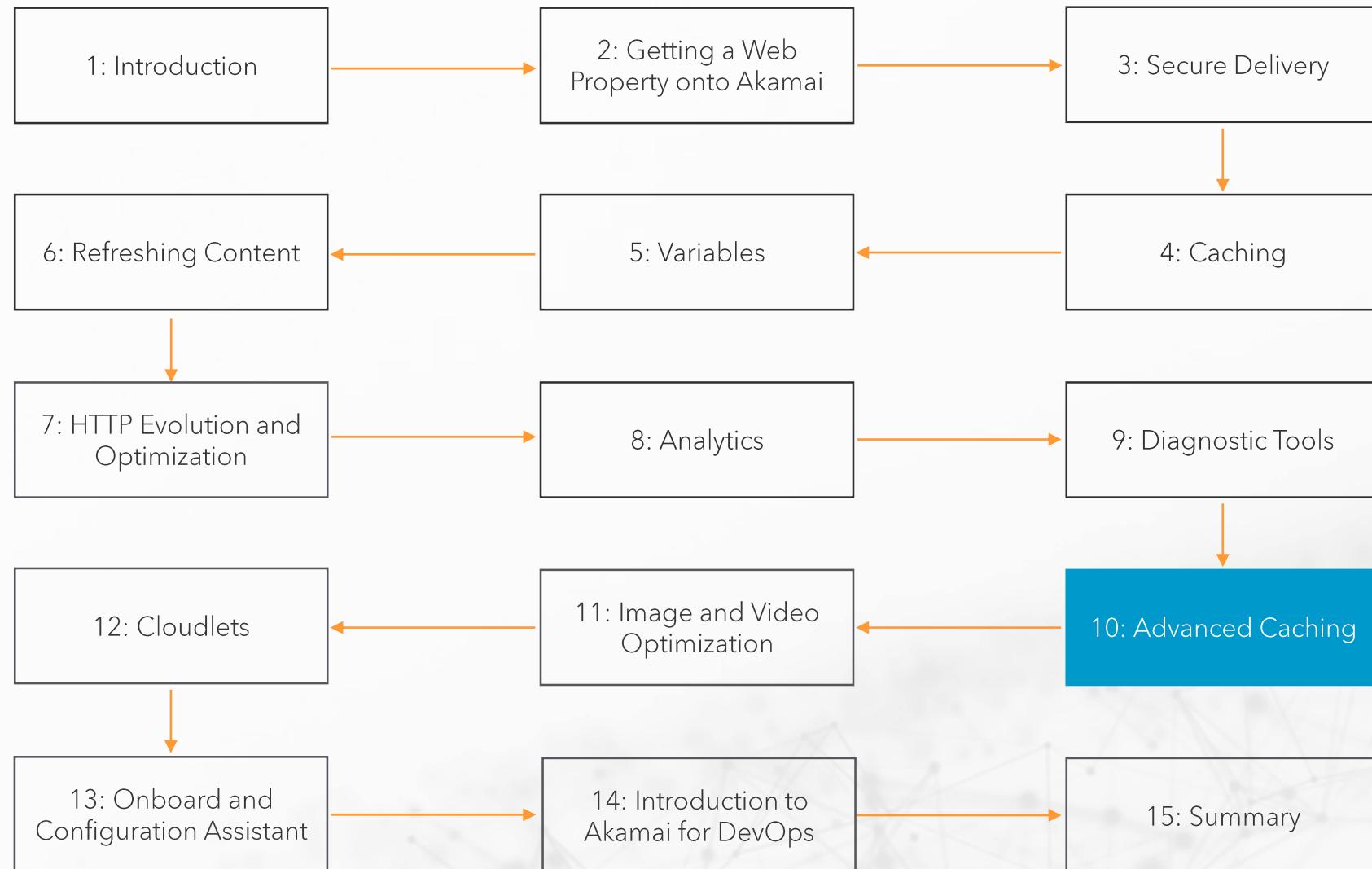
Diagnostic Tools



Q&A



Agenda



MODULE 10

Advanced Caching



Module Objectives

At the end of this module, you will understand:

- The concept of a Cache Key on Akamai.
- How the Cache Key can be manipulated.
- Advanced caching use-cases / examples.
- Nuances involved with query strings and caching.
- How to control caching on Akamai from Origin.



Basic Caching Review

Caching at the Edge

- Increases: **performance** by decreasing time to download objects, **origin offload** by reducing requests
- Increases **control** – allows you to cache and then purge when your objects (code, etc.) change

Caching at the Browser

- You can control caching at the browser in your Akamai property (configuration)
- Be careful because you can't purge browser cache

Cache Key

- **Unique** identifier of content in Akamai Edge server

Purging / Invalidation

- Required for more flexible caching strategies

Advanced Caching - Review

All what was said for Basic Caching section applies to Advanced Caching scenarios. These scenarios are mainly defined by a need to alter a cache key definition.

Cache Key

- Prevents cache proliferation or pollution
- Extremely important to make purging work

Purging / Invalidation

- You need to be able to purge as granular as required

Default Cache Key - Recap

- Add Akamai Pragma headers to retrieve cache key
- Debug information added to response at the Edge

Example:

X-Cache: **TCP_MEM_HIT** from a**23-50-55-14**.deploy.akamaitechnologies.com
(AkamaiGHost/9.6.0-24900238) (-)

X-Cache-Key:
/L/468/315249/30d/demo5.www.wp1.akamaiuweb.com/styles.css

X-True-Cache-Key:
/L/demo5.www.wp1.akamaiuweb.com/styles.css vcd=8080

X-Check-Cacheable: **YES**

```
curl http://demo5.www.wp1.akamaiuweb.com/styles.css -H "Pragma: akamai-x-cache-on,akamai-x-cache-remote-on,akamai-x-check-cacheable,akamai-x-get-cache-key,akamai-x-get-true-cache-key" -vs -o /dev/null --stderr - | grep -E "X-Cache-Key:|X-True|Cacheable|X-Cache:"  
< X-Cache: TCP_MEM_HIT from a23-50-55-14.deploy.akamaitechnologies.com  
(AkamaiGHost/9.6.0-24900238) (-)  
< X-Cache-Key:  
/L/468/315249/30d/demo5.www.wp1.akamaiuweb.com/styles.css  
< X-True-Cache-Key:  
/L/demo5.www.wp1.akamaiuweb.com/styles.css  
vcd=8080  
< X-Check-Cacheable: YES
```

Purging Content from Edge Cache - Recap

- Allows you to cache with long TTLs and then purge (delete) or invalidate content when required
- Can cache by URL, CP Code, Cache Tag, directory, etc.
- Increases origin offload

Tools for Refreshing Content (Purging/Invalidating)

- Fast Purge (CCUv3)
- {OPEN} APIs (<https://developer.akamai.com/api/purge/ccu/overview.html>)
- Control Center UI

* Content Control Utility (CCUv2 soon will be deprecated)

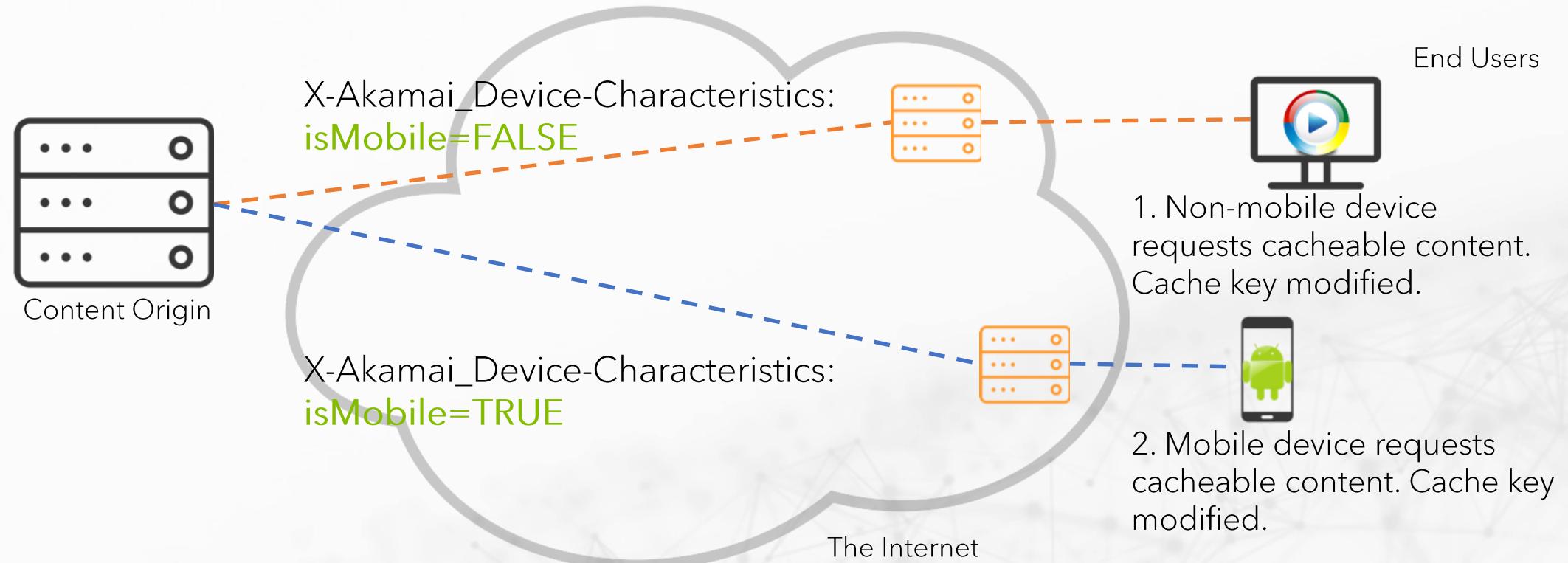


Advanced Caching - What It Means

Example scenario:

Customer wants to cache different HTML for a desktop browser vs mobile

- Cache key will be modified on Edge
- Origin will have to be informed about criteria for cache key



Cache Key Modification - cid

Modification affects cache key:

- creates cid= section
- query string parameters are not appended to path

In our example cache key modification is done based on Device Characterization feature, which adds EDC_IS_MOBILE internal variable to the cache key. This feature works based on User-Agent header value.

Non-Mobile - User-Agent: Windows

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/debug.php vcd=4122
cid= EDC_IS_MOBILE=false

Mobile - User-Agent: iPhone

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/debug.php vcd=4122
cid= EDC_IS_MOBILE=true

Implementation

Cache key

Information to origin

Caching enabled / TTL

Device Characterization - Define Cached Content

Cache Based on These is_mobile

Device Characterization - Forward in Header

Send These Characteristics is_mobile

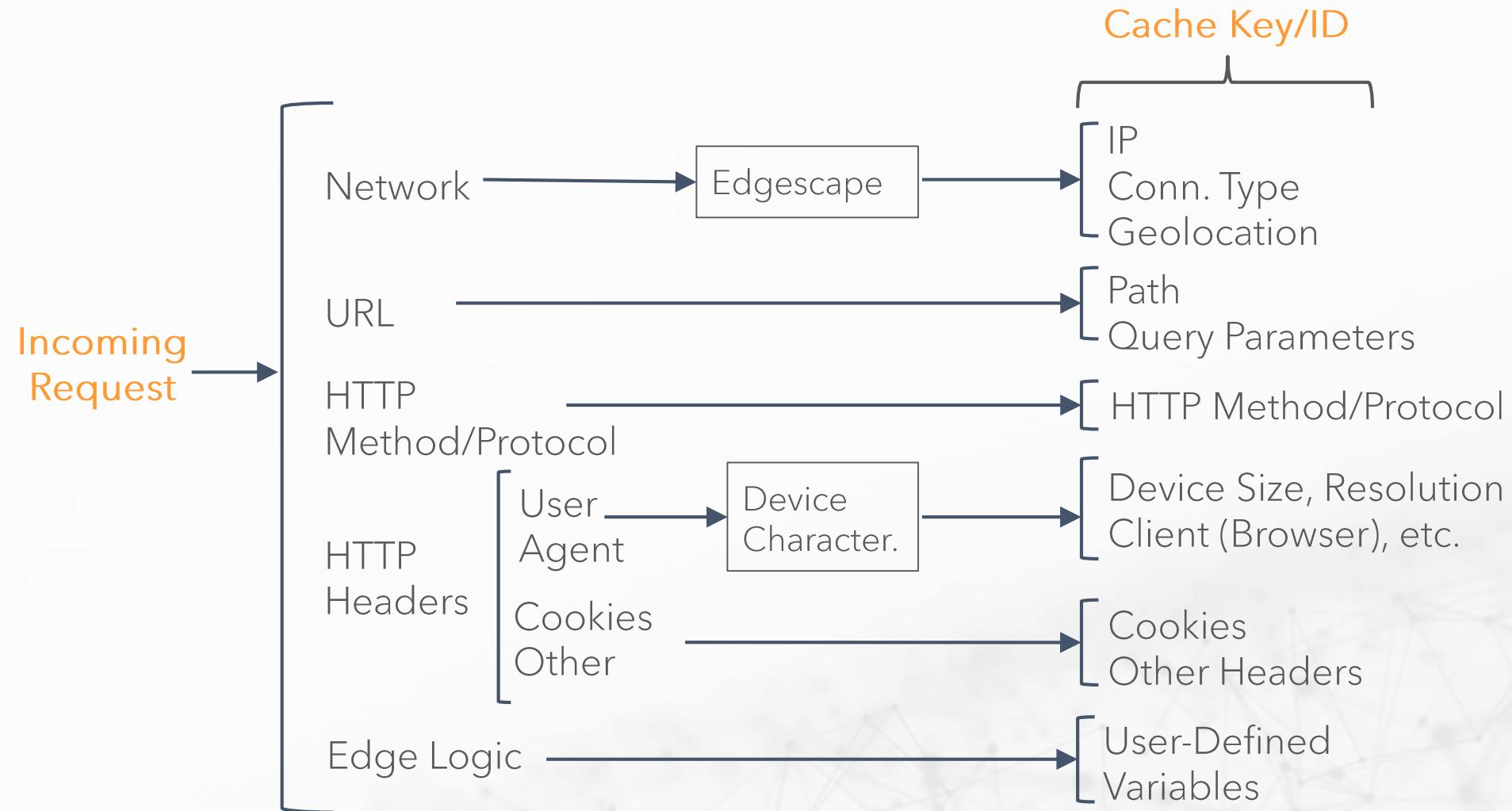
Caching

Caching Option Cache

Force Revalidation of Stale Objects Serve stale if unable to validate

Max-age seconds

Cache Key Modification - Source



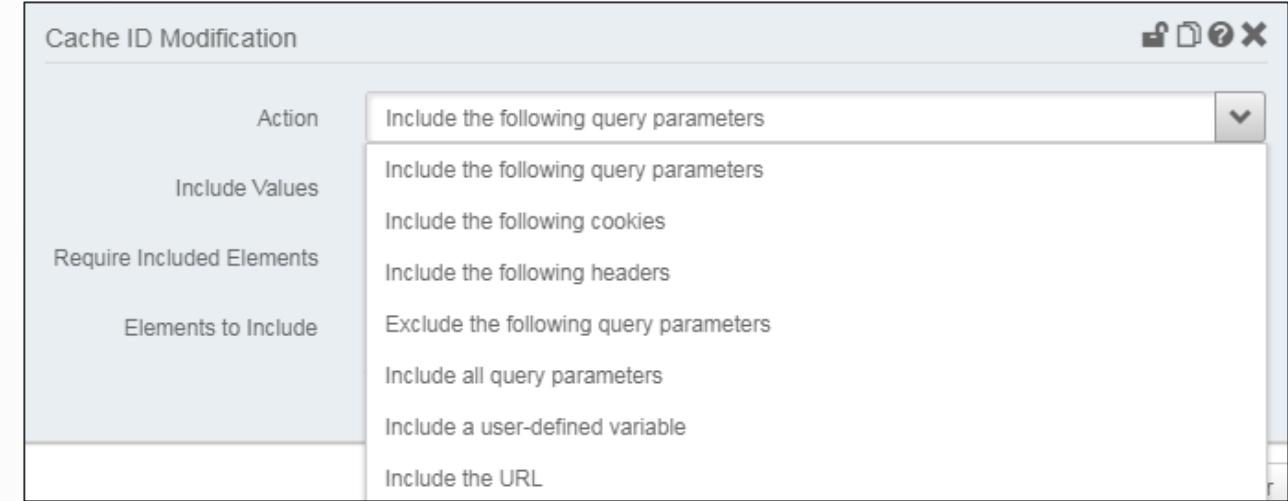
Two Behaviors for Cache ID Modification

Cache ID Modification

Modify Cache ID based on:

- Query String Parameters
- Cookies
- Headers
- User-defined Variables

This behavior can be chained.

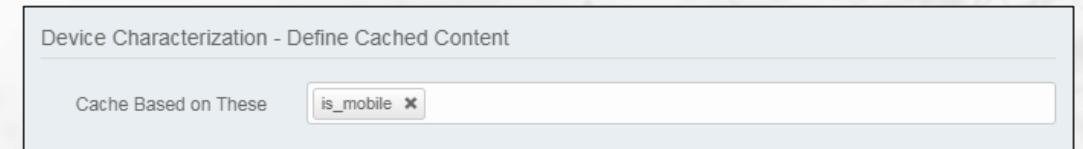


The screenshot shows a configuration interface for 'Cache ID Modification'. On the left, there are three main sections: 'Action', 'Include Values', and 'Require Included Elements'. The 'Action' section is expanded, showing a dropdown menu with the following options:

- Include the following query parameters
- Include the following query parameters
- Include the following cookies
- Include the following headers
- Exclude the following query parameters
- Include all query parameters
- Include a user-defined variable
- Include the URL

Device Characterization - Define Cached Content

Modify Cache ID based on Device Characterization data



The screenshot shows a configuration interface for 'Device Characterization - Define Cached Content'. It has a single input field labeled 'Cache Based on These' containing the value 'is_mobile'.

Additional Use Cases

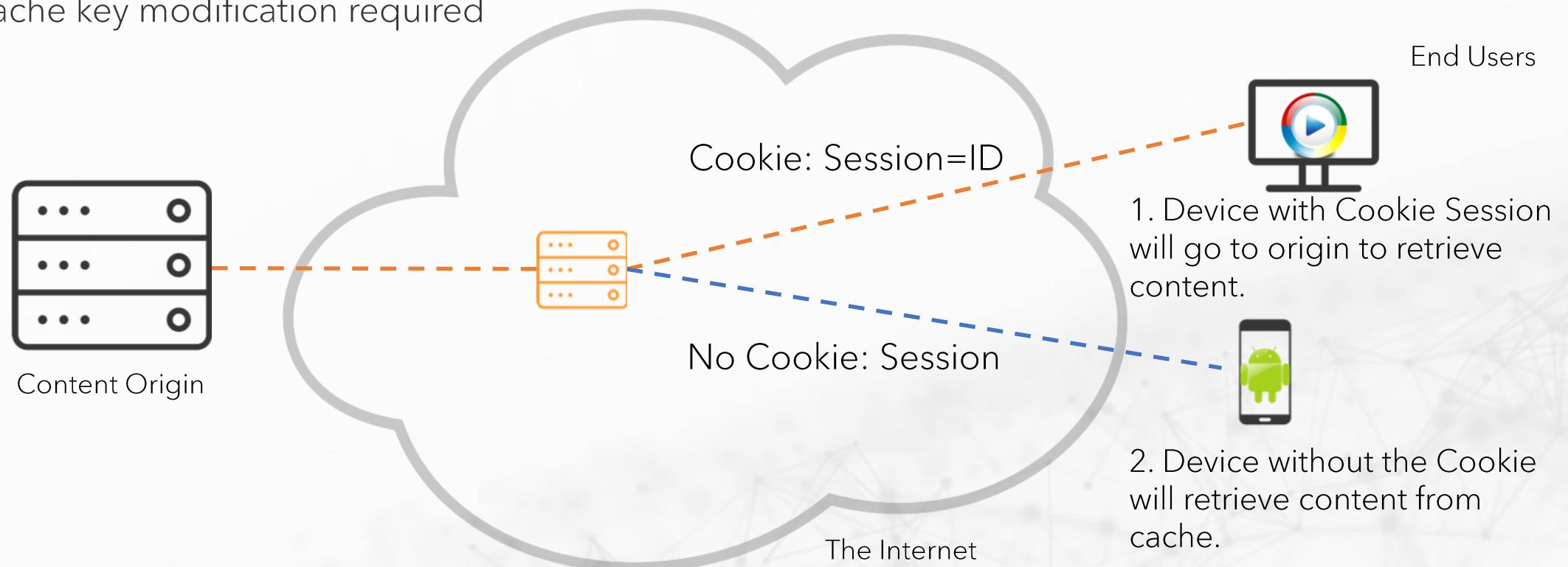
Additional scenarios of Advanced Caching include:

- Cache generic page, bypass page for users with session
- Cache different versions of a page for different locations
- Anything you want to put in a User-Defined Variable
- Caching APIs
- Etc.

Question: Do you have any other scenarios?

Cache Generic and Bypass Cache for Session

- Serve generic (no session/no-login) users from cache
- Bypass cache for users with a session cookie
- No cache key modification required



Cache Generic and Bypass Cache for Session

The default cache key is not modified.

This behavior is achieved by using the “bypass cache” option which will instruct the edge server go forward to the origin for the content even if the requested resource is in cache.

Request does not contain session cookie

```
X-Cache: TCP_HIT from a23-50-55-21.deploy.akamaitechnologies.com  
(AkamaiGHost/9.6.2.1-25376270) (-)
```

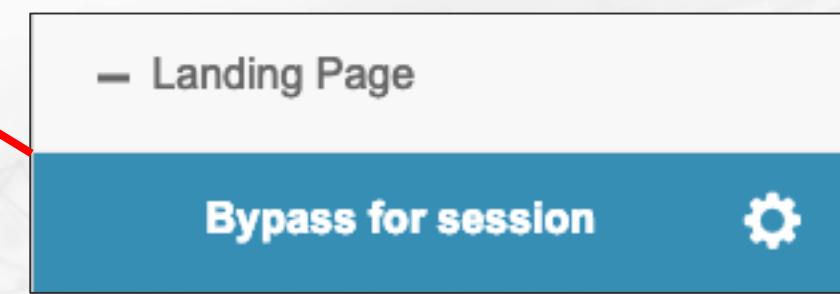
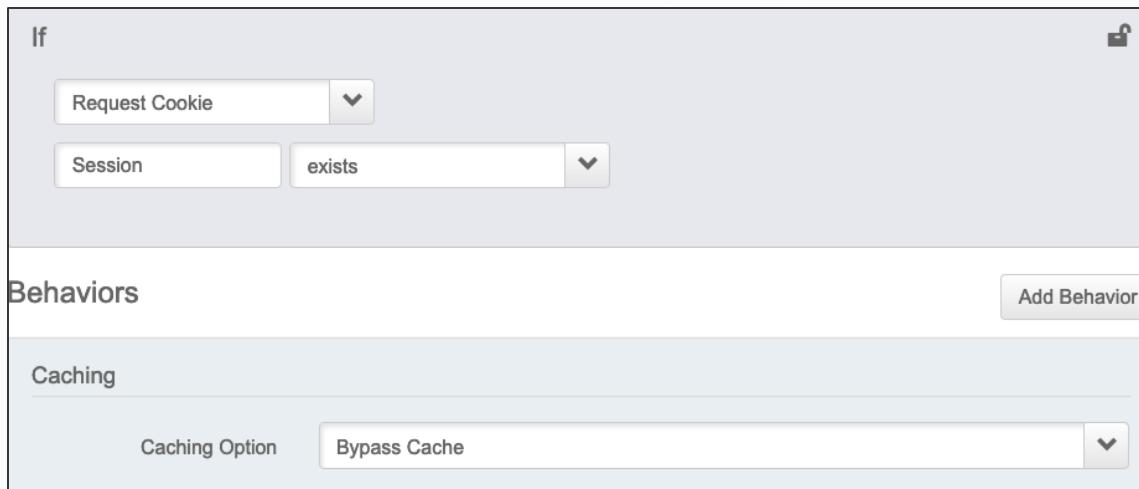
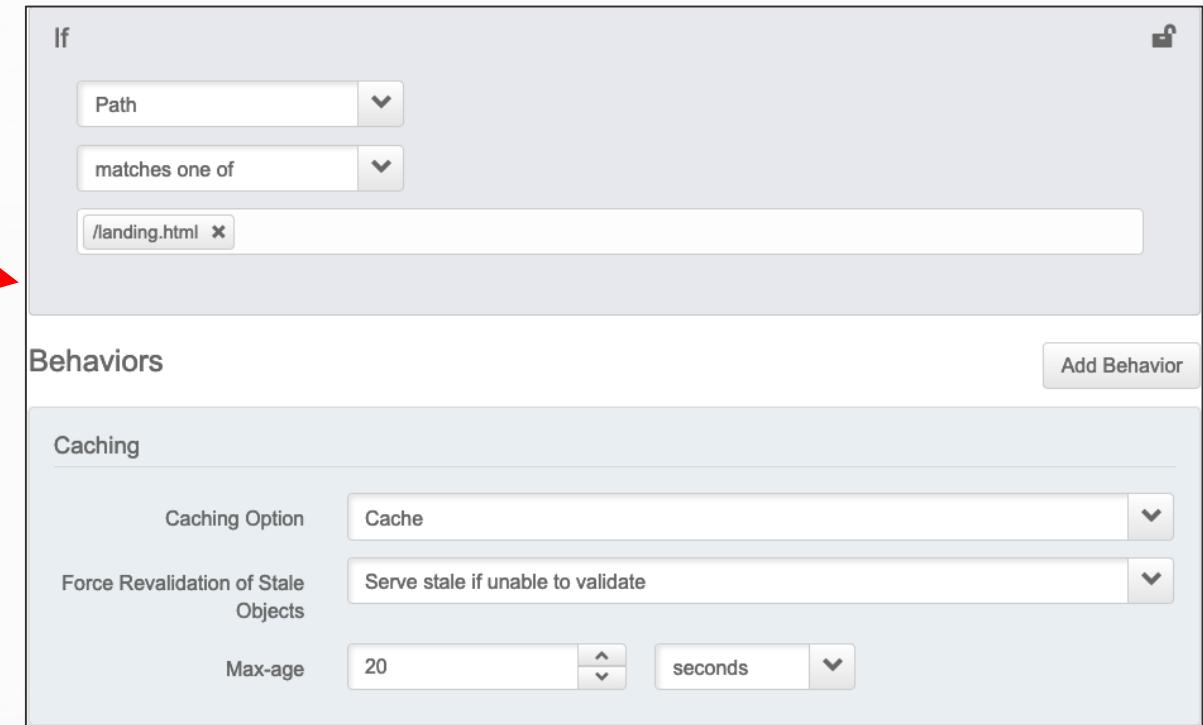
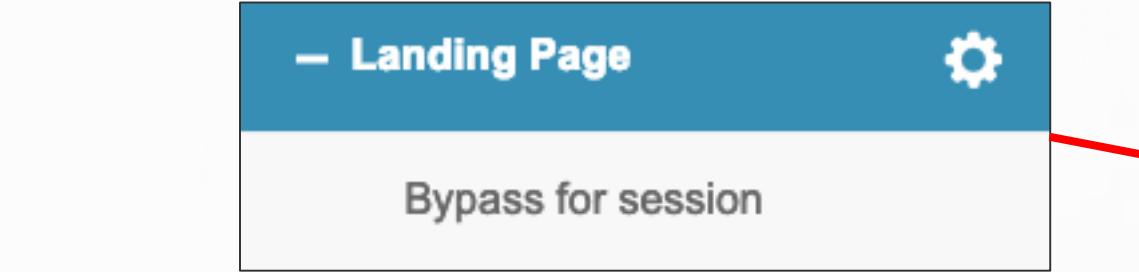
```
X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/landing.html vcd=4122
```

With Session cookie:

```
X-Cache: TCP_MISS from a23-50-55-21.deploy.akamaitechnologies.com  
(AkamaiGHost/9.6.2.1-25376270) (-)
```

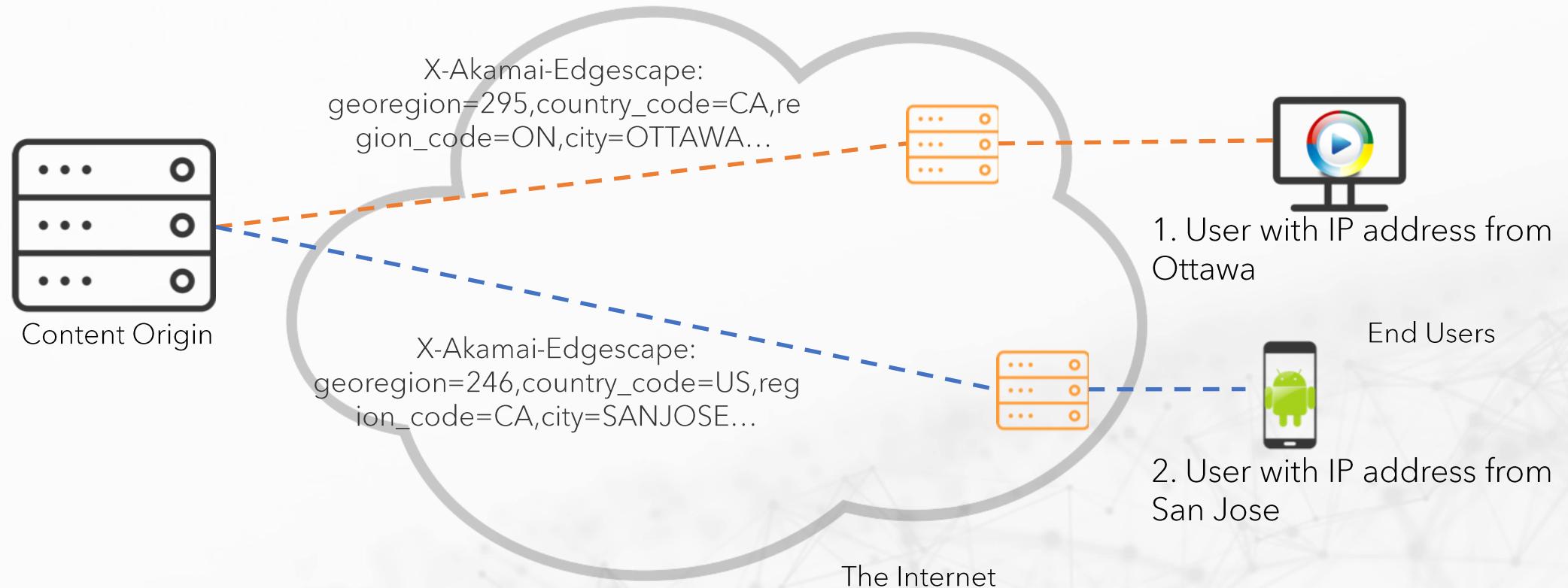
```
X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/landing.html vcd=4122
```

Cache Generic - Implementation



Different Locations

Include information derived from IP addresses of: clients, network gateways, routers; in order to identify users' geographical locations. Cache content accordingly.



Cache Key Modification - cid

Users connecting from the same city will receive the same content since the location is included in the cache key in the form of the variable PMUSER_CITYCODE.

User with Ottawa IP: 184.150.158.20

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/index.php vcd=4122
cid= PMUSER_CITYCODE=OTTAWA

User with San Jose IP: 96.6.45.100

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/index.php vcd=4122
cid= PMUSER_CITYCODE=SANJOSE

Different Locations - Implementation

- Match path as required
- Send location data to origin using Edgescape
- Modify Cache ID to cache location-specific versions
 - Requires using user-defined variables
- Cache it



Different Locations - Implementation

- Match path as required
- Send location data to origin using Edgescape
- Modify Cache ID to cache location-specific versions
 - Requires using User-Defined Variables
- Cache it

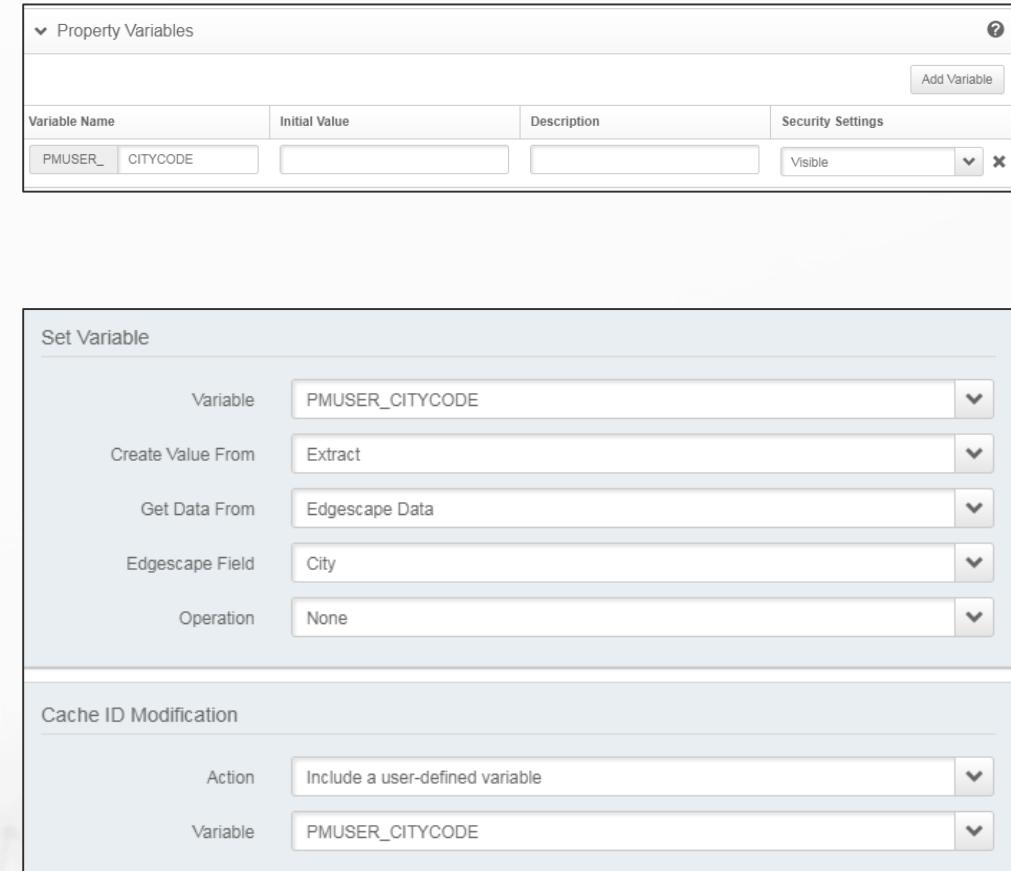


Header Sent to Origin:

X-Akamai-Edgescape: georegion=295,country_code=CA,region_code=ON,city=OTTAWA,lat=45.42,long=-75.70,timezone=EST,zip=K1A+K1B+K1G+K1H+K1J+K1K+K1L+K1M+K1N+K1P+K1R+K1S+K1T+K1V+K1X+K1Y+K1Z+K2A+K2B+K2C+K2E+K2G+K2H+K2P,continent=NA,throughput=vhigh,bw=5000,asnum=174,location_id=0

Different Locations - Implementation

- Match path as required
- Send location data to origin using Edgescape
- **Modify Cache ID to cache location-specific versions**
 - Requires using User-Defined Variables
- Cache it



The screenshot shows two configuration panels in the Akamai Edge Editor:

- Property Variables** (Top Panel):

| Variable Name | Initial Value | Description | Security Settings |
|-----------------|---------------|-------------|-------------------|
| PMUSER_CITYCODE | | | Visible |
- Set Variable** (Bottom Panel):

| | |
|-------------------|-----------------|
| Variable | PMUSER_CITYCODE |
| Create Value From | Extract |
| Get Data From | Edgescape Data |
| Edgescape Field | City |
| Operation | None |
- Cache ID Modification** (Bottom Panel):

| | |
|----------|---------------------------------|
| Action | Include a user-defined variable |
| Variable | PMUSER_CITYCODE |

Different Locations - Implementation

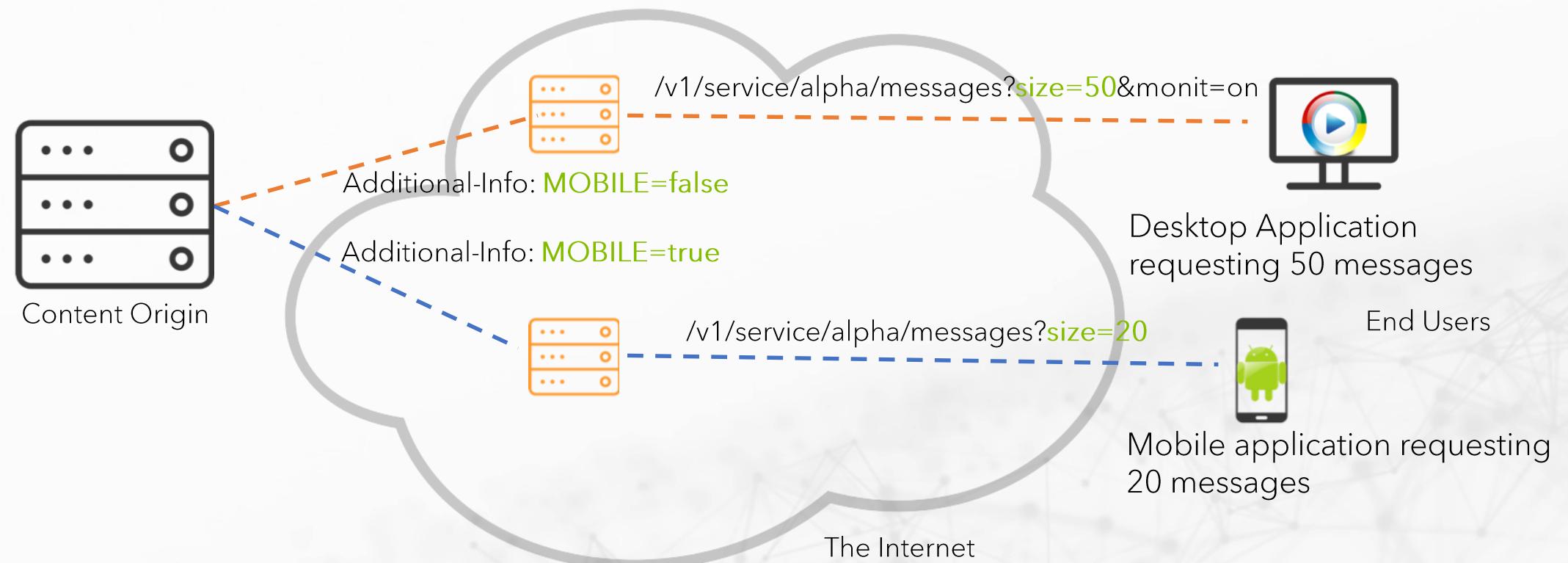
- Match path as required
- Send location data to origin using Edgescape
- Modify Cache ID to cache location-specific versions
 - Requires using User-Defined Variables
- Cache it

Caching

| | |
|-------------------------------------|-----------------------------------|
| Caching Option | Cache |
| Force Revalidation of Stale Objects | Serve stale if unable to validate |
| Max-age | 1 days |

API Caching

REST architected web services are great candidates for caching. We can enhance cacheability by additional information to cache key. In this case we will take an example of GET requests for messages. We will consider the client type (mobile/non-mobile) and query string parameter size.



Cache Key Modification - cid

API calls are cached based on URL and additional parameter from user variable: PMUSER_DEVICE_TYPE_MOBILE which is derived from User-Agent header.

URL: /v1/service/alpha/messages?size=50&monit=on and User-Agent: Windows

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/v1/service/alpha/messages vcd=4122
cid=size=50 PMUSER_DEVICE_TYPE_MOBILE=false

URL: /v1/service/alpha/messages and User-Agent: iPhone

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/v1/service/alpha/messages vcd=4122
cid=size=20 PMUSER_DEVICE_TYPE_MOBILE=true

Different Locations - Implementation

- Match path as required
- Send Device Type information to origin in Additional-Info header.
- **Modify Cache ID to cache by device type and include size query string parameter.**
- Cache it

Cache ID Modification

| | |
|----------|---------------------------------|
| Action | Include a user-defined variable |
| Variable | PMUSER_DEVICE_TYPE_MOBILE |

Cache ID Modification

| | |
|---------------------------|--|
| Action | Include the following query parameters |
| Include Values | Yes |
| Require Included Elements | Not required for caching |
| Elements to Include | size |

Query Strings Parameters and Caching

Default Cache Key - query string parameters:

- All query string parameters are by default part of the Cache Key

Example:

URL: /product.php?productID=2134223&country=DE

X-True-Cache-Key:

/L/demo.aka.akamaiuweb.com/product.php?productID=2134223&country=DE vcd=4122

After Cache Key modification - query string parameters:

- No query string parameters are included by default, you need to enable it.

Example: enabled for productID

URL: /product.php?productID=2134223&country=DE

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/product.php vcd=4122
cid=productID=2134223__

Query Strings Parameters and Caching

What if you have 10,000 products?

- /product.php?productid=33040401
- /product.php?productid=35123002
- /product.php?productid=20022303
- ...
- /product.php?productid=21342223

Problem: How do you Purge all of them?

They are all cached under a separate cache-key.

- Purge using Fast Purge
 - Upside: fast
 - Downside: 10,000 URLs to specify and you need to know them.
- Purge with ECCU
 - Upside: easy to manage 10,000 URLs (just use a wildcard, do not need to know all of them)
 - Downside: slow
- Another solution is to modify the cache key.

Query Strings Parameters and Caching

Moving Query String Parameters to the Cache ID will solve the problem with listing

After the change cache keys looks like:

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/product.php vcd=4122 cid=productID=21342223__

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/product.php vcd=4122 cid=productID=20022303__

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/product.php vcd=4122 cid=productID=33040401__

How do you purge now?

- Use Fast Purge, and specify one URL: /product.php
- This will purge all **cid=** variants

Cache ID Modification

| | | |
|--------|------------------------------|---|
| Action | Include all query parameters | ▼ |
|--------|------------------------------|---|

Query String Parameter Order

Sometimes when you have multiple QSPs, the order can get mixed up.

- /products.php?productID=33040342&lang=DE

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/products.php vcd=4122
cid=productID=33040342&lang=DE

- /products.php?lang=DE&productID=33040342

X-True-Cache-Key: /L/demo.aka.akamaiuweb.com/products.php vcd=4122
cid=lang=DE&productID=33040342

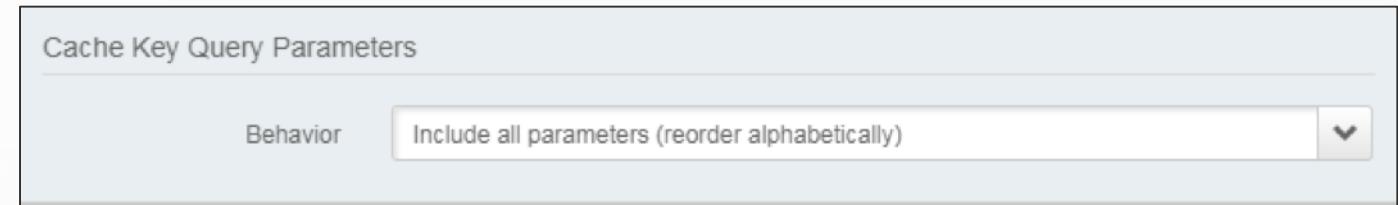
Are these pages the same? - Most likely yes.

How does it affect caching? - They are unique items in cache therefore diluting your cache.

Query String Parameter Order

Re-ordering Query String Parameters

- Cache hit-rate increase
- Origin offload increases
- Easy to purge all page variants



Pages

- /products.php?productID=33040342&lang=DE
- /products.php?lang=DE&productID=33040342



Map to this single Cache Key

X-True-Cache-Key:
/L/demo.aka.akamaiuweb.com/products.php vcd=4122
cid=lang=DE&productID=33040342

Unnecessary Query String Parameters

What about random Query String Parameters (QSP) your URLs?

- QSPs added by third-party analytics tools, etc.
- /product.php?productID=001&ref=fb, where **ref=fb** is a query string that represent client-side functionality or is not used in content generation on origin

Remove them from the Cache ID:

Use Cache ID Modification to include only your desired QSPs in the Cache ID

Cache ID Modification

| | |
|---------------------------|--|
| Action | Include the following query parameters |
| Include Values | Yes |
| Require Included Elements | Not required for caching |
| Elements to Include | productID |

Unnecessary Query String Parameters

Removing unnecessary QSPs from the Cache ID:

- Cache hit-rate increase
- Origin offload increases
- Easy to purge all page variants

These pages

```
/product.php?track=true&productID=660513  
427&ref=fb  
/product.php?track=false&productID=66051  
3427&ref=tw  
/product.php?track=false&productID=66051  
3427&ref=fb
```



Map to this single Cache Key

X-True-Cache-Key:
/L/demo.aka.akamaiuweb.com/product.php
vcd=4122 cid=productID=660513427

Controlling Edge Cache from Origin

There are several ways to control your Edge Cache functionality from origin:

- Honor Cache-Control - caching
- Edge-Control Headers - caching override
- Cache Tags from Origin - additional purging capabilities

Controlling Edge Cache from Origin

To use Cache-Control and Expires headers:

- You need to set cache-control headers at origin
- Enable honoring these settings in Edge cache
- This can be limited by match conditions

Caching

| | |
|-------------------------------------|-----------------------------------|
| Caching Option | Honor Origin Cache Control |
| Force Revalidation of Stale Objects | Serve stale if unable to validate |
| Default Max-age | 1 hours |

Honor Origin Cache Control

- Cache
- No Store
- Bypass Cache
- Honor Origin Cache Control and Expires
- Honor Origin Cache Control
- Honor Origin Expires

Controlling Edge Cache from Origin

Edge-Control Headers

- Sent from Origin
- Honored by the Edge regardless of caching configuration
- Override other TTLs/Caching configuration settings
- Edge-Control headers have the same spec as Cache-Control

Example:

```
Edge-Control: !no-store, max-age=20s
```

```
Edge-Control: bypass-cache
```

Cache Tagging

- Objects in Cache can be tagged for purging
- Done by adding [Edge-Cache-Tag](#) header to response
 - Done at Origin
 - Done at Edge (using [Modify Incoming Response Header](#) behavior)
 - Supports multiple tag (more than one dimension)
- Fast Purge supports purge by tags

Example:

Edge-Content-Tag: jackets, blue

Edge-Content-Tag: blouse, rose

Cache Tagging - Edge

At the Edge using [Modify Incoming Response Header](#) behavior

Modify Incoming Response Header

| | |
|--|---|
| Action | <input type="button" value="Add"/> |
| Select Header Name | <input type="button" value="Other..."/> |
| Custom Header Name
(variable support) | <input type="text" value="Edge-Content-Tag"/> |
| Header Value
(variable support) | <input type="text" value="product-page,jackets"/> |

LAB 5

Advanced Caching & Device Characterization



Q&A



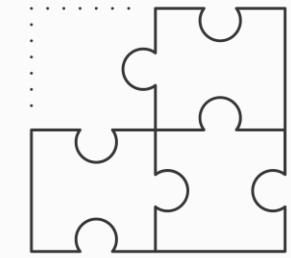
Summary

Review of Basic Caching

- Caching at the Edge
- Caching at the Browser

Query String Nuances

- Moving Query String Parameters to the Cache ID
- Removing Query String Parameters
- Re-ordering Query String Parameters
- Controlling Caching from Origin



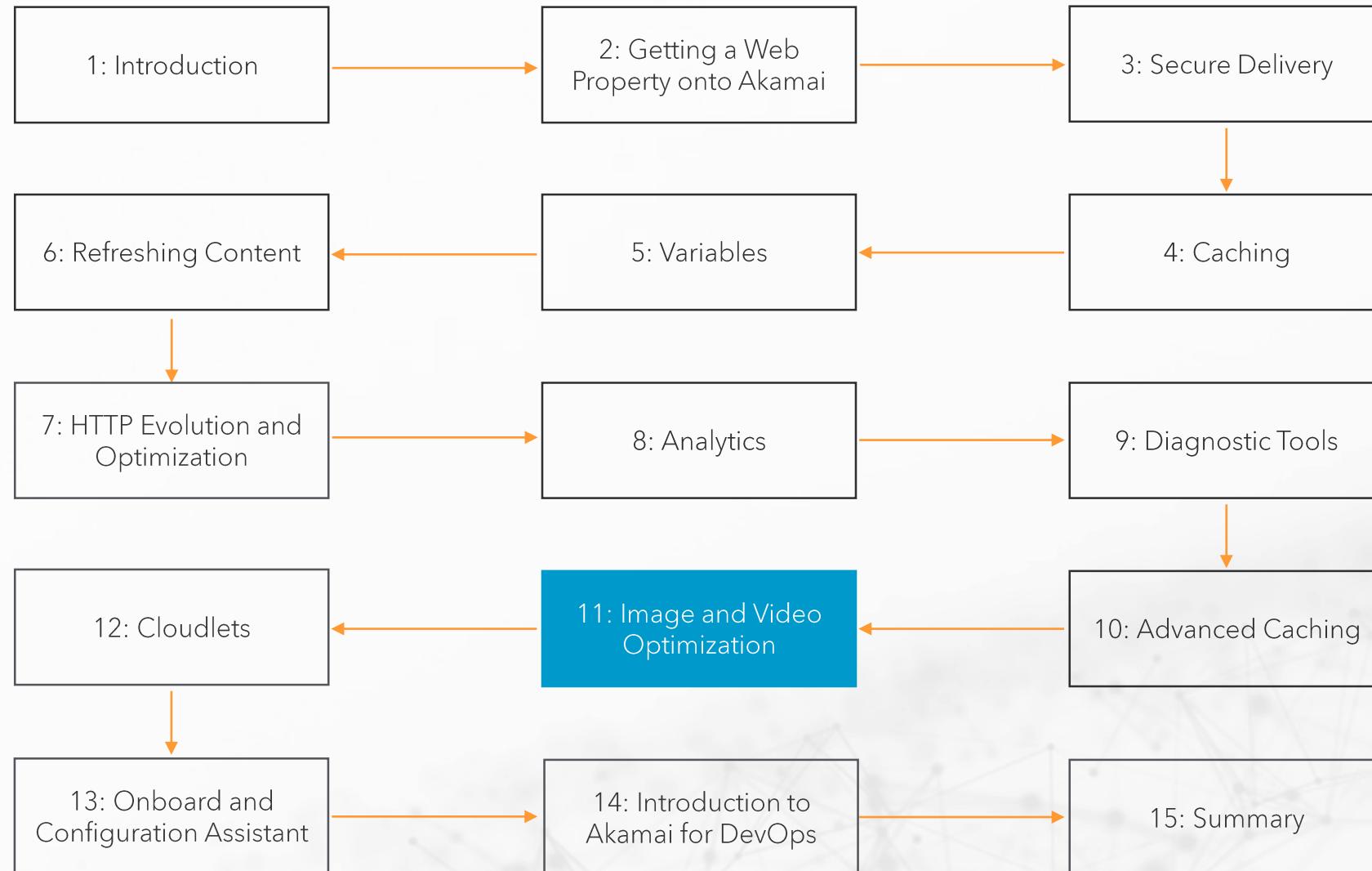
Advanced Caching

- Cache ID Modification
- Advanced Caching Use-Cases
 - Cookie (or Session) vs no cookie/session
 - Different Locations
 - User-Defined Variables
 - And many more...

Controlling Caching from Origin

- Honor cache-control
- Edge-control headers
- Cache Tags

Agenda



MODULE 11

Image and Video Optimization



Module Objectives

At the end of this module, you will understand:

- Challenges of delivering images online.
- Image Management.
- Adaptive Image Compression.



Challenges of Images Online

Key challenges in delivering images online include:

- Demand for image-rich web experiences
- Different devices and browsers
- Over-downloading: downloading images of excess resolution/quality

Challenges - Demand for Image-Rich Web Experience

End-users have come to expect image-rich web experience and, of course, they also expect it fast!
Images are about 65% of total page weight in bytes.

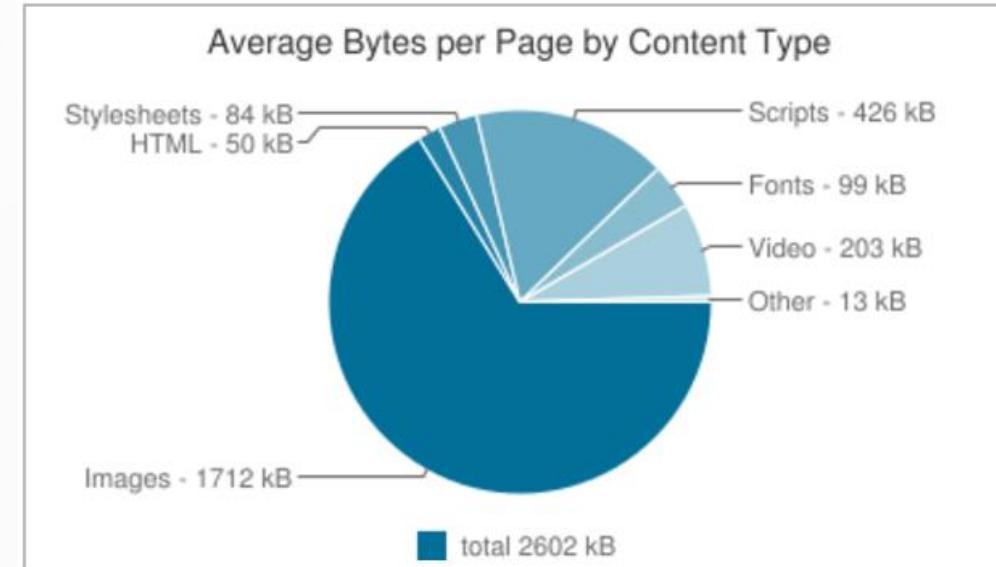


Chart from httparchive.org (April 2017)

Challenges - Different Devices and Browsers

End-users are now using all kinds of devices and browsers.

Device Types include many variations of:

- Mobile - Androids, iPhones, Windows Phones, Sony, etc...
- Desktop/laptop devices
- Tablets - iPad/Surface/Android tablets
- IoT - Internet of Things

Browsers include

- Safari
- Chrome
- Edge / IE
- Firefox
- Etc.

Number of Devices x Number of Browsers
= a very large number of scenarios

Many different screen sizes and image support issues

Challenges - Browser-Optimized Image Formats

Browser-optimized image formats offer superior performance, but only on specific browsers.

For example:

- Safari - supports JP2 (JPEG 2000)
- Chrome - supports WebP
- IE 11 / Edge - JPEG XR

Tip

<http://caniuse.com/> can be helpful in determining browser support for different things, including image formats.

Browser-optimized formats can save a lot of bytes and increase rendering speed, but you need to know when you can use them.

Challenges - Over-Downloading

Serving images that are big enough for larger-screen devices often results in smaller devices (ex iPhones, etc) downloading images of excessive size and quality.

Responsive Web Design (RWD) site:

- Serves all devices from a single hostname
- Same content is sent for all devices
- Browser decides locally how to render the content - this includes resizing images to appropriate sizes within the browser

Example:

Image on Desktop

```

```

Same Image on Mobile

```

```

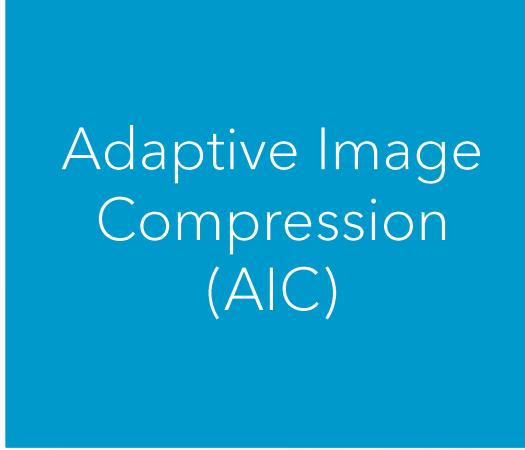
This is an example of over-downloading and it is very common!

Akamai Image Management Solutions

Akamai has several image-related products / features, including:

A solid blue rectangular box containing the text "Image Manager".

Image Manager

A solid blue rectangular box containing the text "Adaptive Image Compression (AIC)".

Adaptive Image
Compression
(AIC)

Image Manager - Introduction

Image Manager helps by leveraging capabilities listed below:

- Creating a variety of optimized derivative images of various sizes and image formats
- Knowing about different device/browser types
- Applying a perceptual-quality based compression across different image formats
- Serving the right derivative image in the right scenario
- Agility by introducing variables into policy definitions
- Optimizing videos to fit user's device characteristics
- Customizable HTML5 Media Viewer for images and videos
- API support for policy modifications

Features: Scale for Mobile and Use Best File Type

The [Scale for Mobile](#) feature automatically resizes delivered image:

- Creates derivative images with different widths, as defined in the IM Policy
- Automatically serves the optimal image width based on
 - Device width as determined by the user-agent and Akamai Device Characterization
 - The **imwidth** query string parameter
 - The Width parameter in the **Image Manager: Set Parameter** behavior

IM chooses the next-size-up image breakpoint.

The [Use Best File Type](#) feature:

- Creates derivative images in different image formats (JPG, WebP, JP2, etc.)
- Automatically services the best image format that the device/browser can use

Features: Scale for Mobile and Use Best File Type

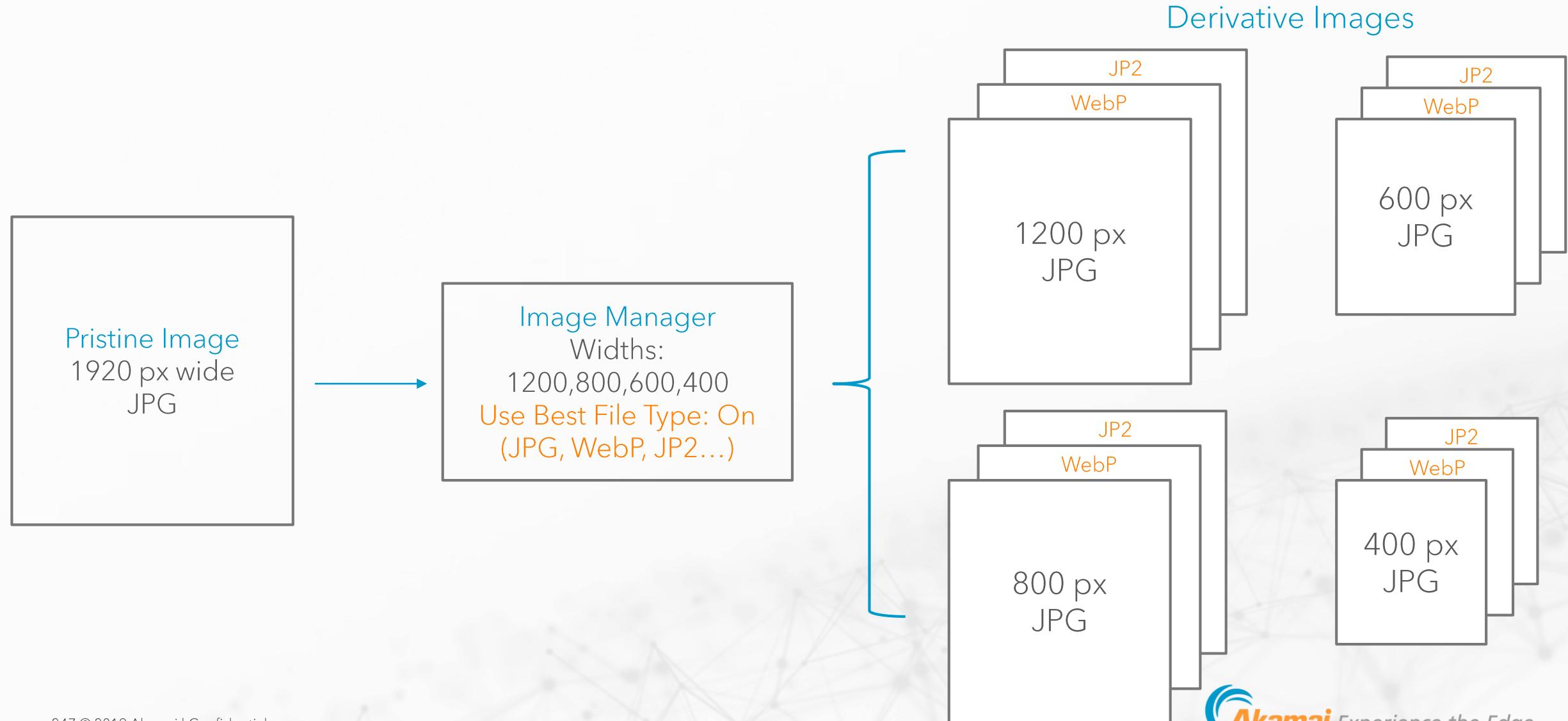


Image Manager - Policies

Image Manager *Policies* are where you define:

- Image widths (breakpoints)
- Compression settings
- Additional transformations (ex greyscale, image overlay, etc.)

By default, a single policy exists, called the *Default Policy*, which:

- Is also called the .auto policy in debug headers.
- Has default image widths (breakpoints) 5000, 2048, 1024, 640, 320
- Defaults compression to encoding level 90
- No transformations

Image Manager - Parameters

Other Query String Parameters

In addition to the **imwidth** query string parameter, Image Manager also supports other parameters:

- **impolicy** - allows you to specify an alternate policy (i.e. beside the default). Both the default and alternate policies are configurable in Policy Manager. Example:

```

```

- **imformat** - allows you to manually specify a specific output image format. Usually it is best to just let Image Manager do this for you automatically. Example:

```

```

Image Manager - Policy Variables

Image Manager policies also supports variables. This allows to parametrize the transformations in policy making them dynamically adjustable.

- **Named** - variable has a name defined in a policy
- **Default value** - initial value when there are no incoming values
- **Referenceable** - can be referenced by query string in URL or set in configuration

This feature allows you to dynamically parametrize IM policy behavior while leveraging all other benefits.

```
  

```

Image Manager - Video

Image Manager supports also the video transformations. Use cases including short videos are supported.

- **Progressive download** – delivery of videos
- **Origin agnostic** – resources are pulled from origin on request and transformed on Akamai Platform.
- **Automated optimization** – depending on device type and browser of:
 - **Right format** – include H.264, H.265 and VP9
 - **Right resolution** – same switch as in images Scale for Mobile
- **Quality selection** on policy level

Image Manager - API

Image Manager policies management can be done also using API calls.

Operations can be performed on:

- Policies - management actions and modifications.
- Images - get images connected with policy and their metadata
- Image Collections - grouping to allow to use Image Manager media viewer on your site



Details of API:

https://developer.akamai.com/api/web_performance/image_manager/v2.html#apisummary

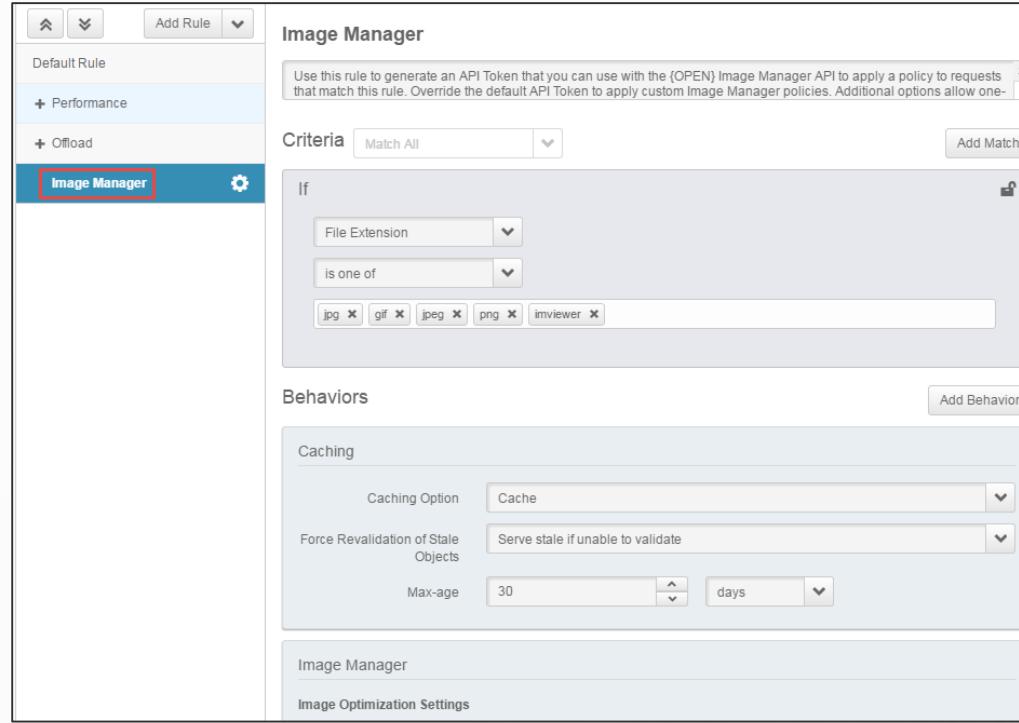
Image Manager Configuration

Image Manager configuration consists of two steps:

1. Add the Image Manager rule in Property Manager
2. Configure your policy (or policies) in Policy Manager

Configuration - Property Manager

The Image Manager rule template will add a new rule with the required match condition and behaviors.



The screenshot shows the Akamai Configuration - Property Manager interface. The left sidebar has buttons for 'Default Rule', 'Performance', 'Offload', and 'Image Manager', with 'Image Manager' being the active tab and highlighted with a red box. The main area is titled 'Image Manager' and contains the following sections:

- Criteria:** Set to 'Match All' with an 'Add Match' button.
- If:** A dropdown menu is set to 'File Extension' with a sub-menu 'is one of' containing 'jpg', 'gif', 'jpeg', 'png', and 'imviewer'.
- Behaviors:** An 'Add Behavior' button is present.
- Caching:** Caching Option is set to 'Cache', Force Revalidation of Stale Objects is set to 'Serve stale if unable to validate', and Max-age is set to '30 days'.
- Image Manager:** An 'Add Rule' button is present.
- Image Optimization Settings:** A section for configuring image optimization settings.

Match Condition

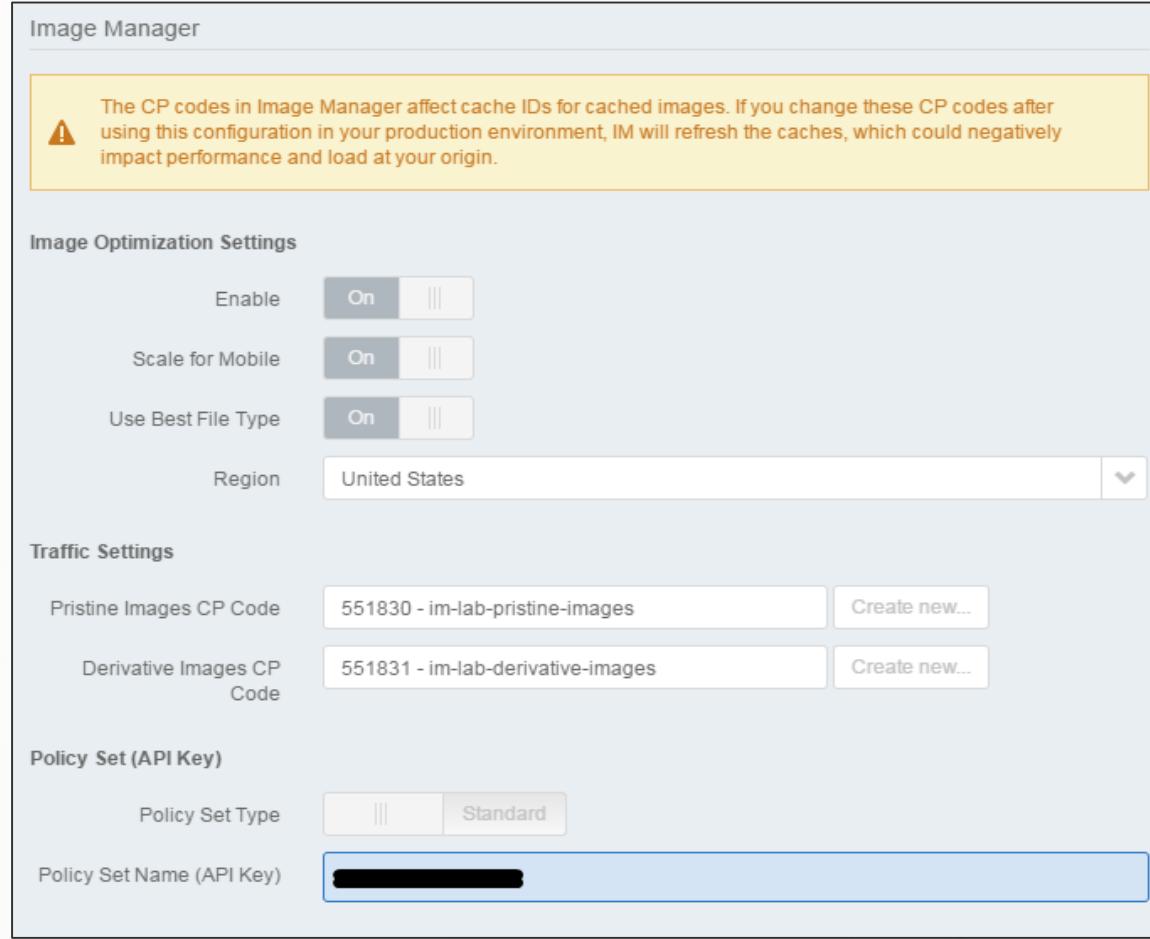
By default this includes a variety of common image file extensions. You may want to add to this list.

If required, you can also add a path match to match only on images in a specific path.

Caching

Image Manager requires that images be cacheable for at least 1 day. By default, this rule template sets the caching to 30 days.

Configuration - Image Manager Behavior



The screenshot shows the 'Image Manager' configuration interface. The 'Image Optimization Settings' section contains three toggle switches: 'Enable' (On), 'Scale for Mobile' (On), and 'Use Best File Type' (On). Below these is a 'Region' dropdown set to 'United States'. The 'Traffic Settings' section contains two text input fields: 'Pristine Images CP Code' (551830 - im-lab-pristine-images) and 'Derivative Images CP Code' (551831 - im-lab-derivative-images), each with a 'Create new...' button. The 'Policy Set (API Key)' section shows 'Policy Set Type' as 'Standard' and 'Policy Set Name (API Key)' as a redacted field.

Image Optimization Settings

The [Scale for Mobile](#) option is off by default. Make sure you set this to On if you want to take advantage of Image Manager's automatic resizing.

[Use Best File Type](#) will serve the best image format that the browser can work with.

Region

Choose a region where the majority of your end-users are located.

CP Codes

You can use separate CP codes for tracking pristine vs derivative images. CP Codes are primarily used for reporting and billing purposes.

API Token

The API Token is used in Image Manager Policy Manager and/or {OPEN} API, where you can define your Image Manager settings.

Configuration - Policy Manager

Default Policy

In Policy Manager you should see your Default Policy. This comes with default settings and is configurable so you can change your image width breakpoints, compression level and add transformations.



The screenshot shows the Akamai Policy Manager interface. At the top, there is a 'Policy Set Selector' dropdown set to 'Contract' with '1-1NC95D' selected, a 'Policy Set Name' dropdown with '10664568' selected, and an 'Manage Policies' button. Below these are 'Delete Selected Policy Set' and 'Add New Policy' buttons. The main area is titled 'Policies' with a search bar 'Start typing to filter'. It shows a table with columns: 'Custom Policy Name', 'Staging', 'Production', and 'Actions'. A row for 'Default Policy' is shown, with 'Active' in the Staging column and 'Rolling Out' in the Production column. An ellipsis button is in the Actions column.

Clicking into the staging version of the Default Policy allows you to edit it:

- First you edit the staging version of a policy, then you activate it on staging, then you activate it on production (once you're happy with it).
- You can also use the Add New Policy button to create a new policy.

Configuration - Policy Manager: Editing a Policy

After clicking into an existing policy or creating a new one, you can edit the settings.

Image Policy Editor

Policy Name: Default Policy

You are editing this policy

Save and Test on Staging

Derivative Image Quality (defaults 90)

Use Default

Image Quality Level (defaults 90)

Workflow

Background Color

Color (hex) bgcolor

Transformations

Derivative Image Widths (pixels)

5000,2048,1024,640,320

Preview

Update

Transformations

Default Derivative Widths

Configuration - Policy Manager: Editing a Policy

You can enter your own break points (image widths), for example, if you wanted 1200,800,600,400,200 you would just type that in:

Derivative Image Widths (pixels) 

There is a section that allows to define and change policy variables:

| Variable Name | Data Type | Default Value | |
|---------------|-----------|---------------|---|
| bgcolor | color | #FFFFFF |  |
| test | number | |  |

Create Variable

This field is required.

Configuration - Policy Manager: Editing a Policy

By default, Image Manager uses a fixed encoding quality of 90:

Derivative Image Quality ?

Use Default ▾

You can change this to either a "Perceptual Quality", which is recommended, or use a 1-100 scale:

Derivative Image Quality ?

Perceptual Quality ▾

High ▾

Derivative Image Quality ?

Quality Level ▾

80 ▾

Image Manager - Compression

There are two types of image compression:

1. Quality Level

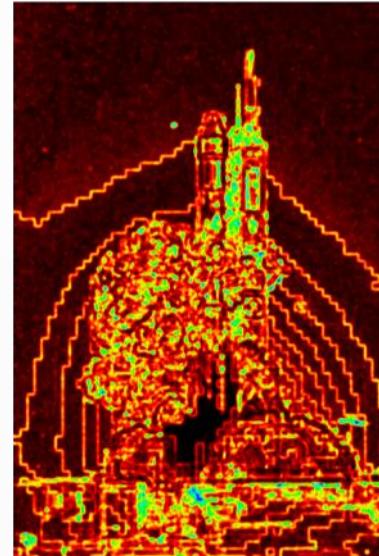
Normalized quality setting across all image supported image formats from 1-100

2. Perceptual Quality

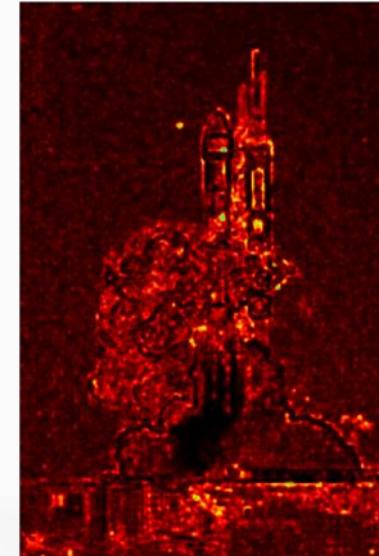
- Intelligently compresses images down to a targeted visual perceptual level
- Uses Structural Similarity (SSIM) algorithm to achieve highest bytes savings while maintaining desired visual integrity
- Perceptual Quality compression with a setting of Medium High is the best practice for most situations

Perceptual Quality Compression

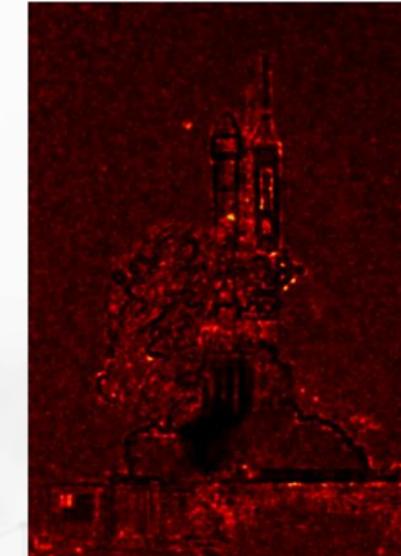
- Leverages [Structural Similarity](#) algorithm
- Structural Similarity generates a human benchmark to compare different image qualities to an original
- [DSSIM](#) - an index of dissimilarity from 0 (perfect match) to 1 (completely different)



Q10
DSSIM: 0.0566



Q50
DSSIM: 0.0251



Q80
DSSIM: 0.0179

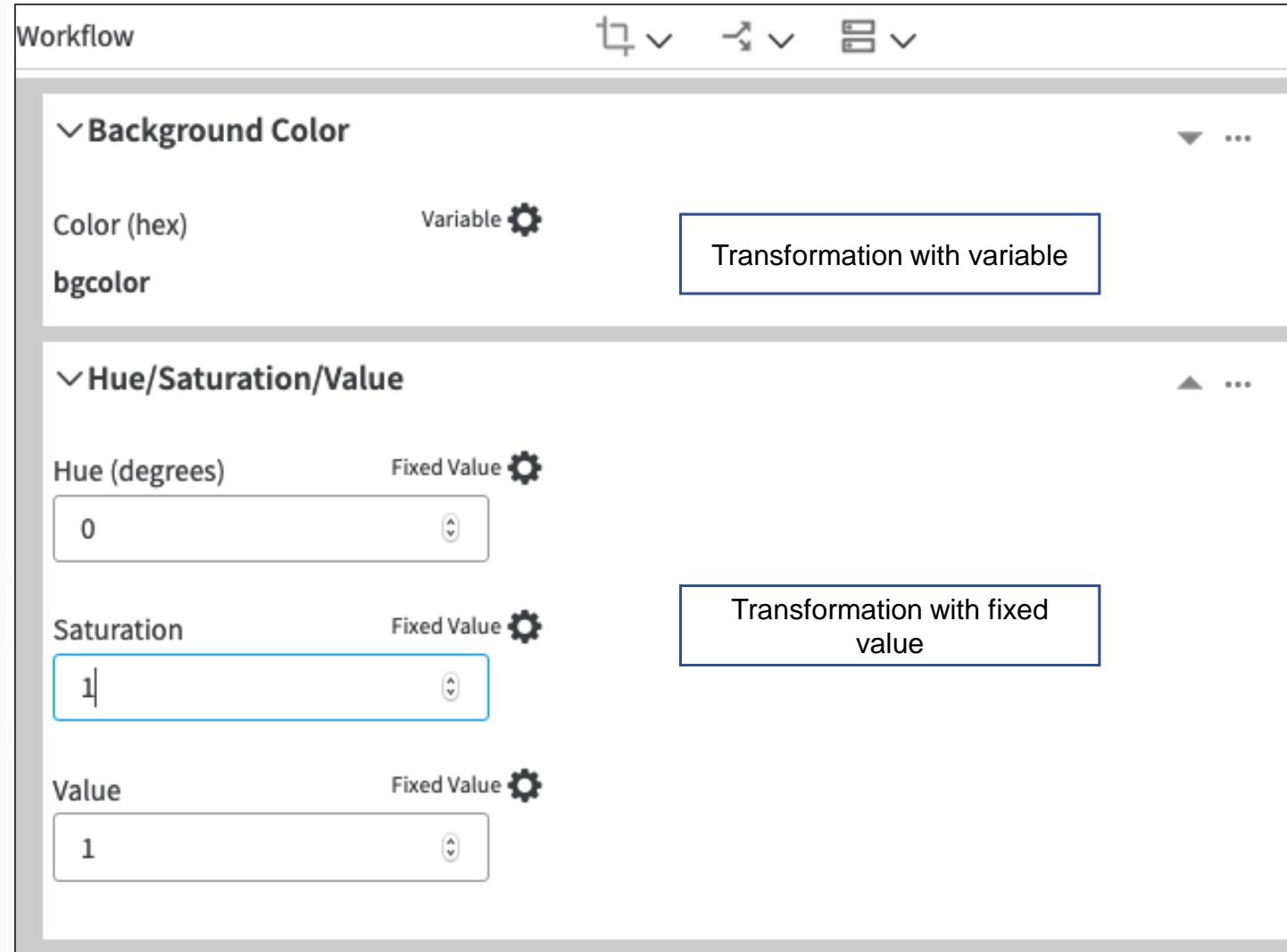
Perceptual Quality - Recommendations

Perceptual Quality Guidelines

| Perceptual Quality | Description |
|------------------------|--|
| Medium High | This is the recommendation for most customers/situations. It is the best balance in terms of bytes saved and visual quality. Visual degradation is very difficult to notice in most normal viewing conditions. |
| High | This would be for exceptionally image quality conscious customers or websites, for example luxury brands may like this setting. |
| Lower than Medium High | At these quality settings, images will have noticeable degradation. For lower-quality settings, it's recommended to use the traditional quality setting (i.e. the 1-100 scale) rather than the perceptual quality. |

Configuration - Policy Manager: Transformations

Workflow



Background Color

Color (hex) Variable 

bgcolor

Transformation with variable

Hue/Saturation/Value

Hue (degrees) Fixed Value 

0

Saturation Fixed Value 

1

Value Fixed Value 

1

Transformation:

- Chained – you can use multiple transformation
- Parameters – exposed as variables

Variables:

- Can be referenced from policies and used as query string parameters
- Must have a default value

Configuration - Policy Manager: Video

Video policies have options to specify:

- Video Quality (Low, Medium, High)
- Video Resolution

■ Video Policy Editor

Policy Name: Default Policy

You are editing this policy

Save and Test on Staging

Derivative Video Quality (②)

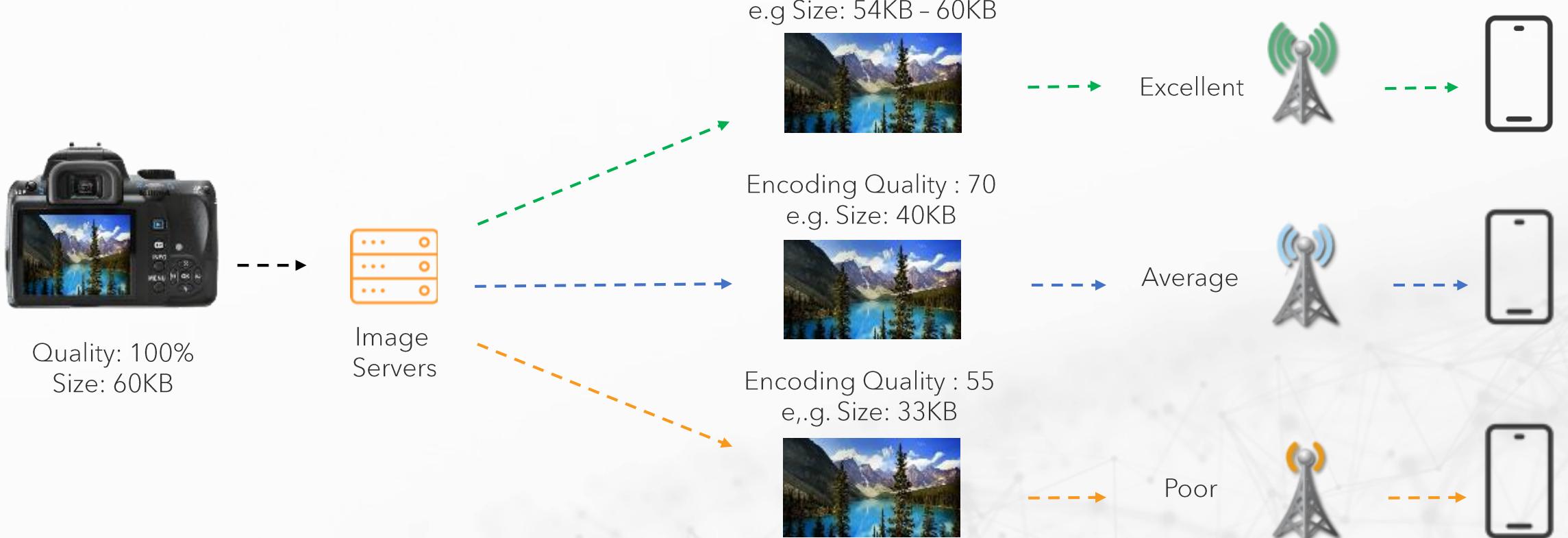
Derivative Video Resolutions (pixels) (②)

- 853 x 640
- 1280 x 720
- 1920 x 1080

Other screens shown previously looks identical to Image - IM configuration.

Adaptive Image Compression (AIC)

AIC modifies compresses JPEG images based on evaluation of the network conditions of the end user's device.



AIC Recommendations: Use Cookies for Testing

You can use cookies during testing to simulate different conditions.

| Cookie | Setting | Description |
|--------------|---------|---|
| aic-bypass | True | Turns off for a request, which is useful when trying to determine whether an issue is at origin or within the edge network. |
| aic-quality | 1–100 | Manually sets quality level, overriding the detected network condition tier. Requests with this cookie will bypass the cache as they do not match a configured setting. |
| aic-network | Mobile | Makes requests appear to come from a mobile network. This is a useful testing method if you have enabled AIC for mobile networks only. |
| aic-rtt | >1 | Uses the value entered (in ms) to override the detected RTT value. (Enter <10 for excellent conditions or > 1500 for poor conditions.) |
| aic-failover | False | Turns off failover. With failover on, it is hard to visually detect web-page issues: When there are problems, AIC will fetch the original image. |

LAB 6

Image Manager



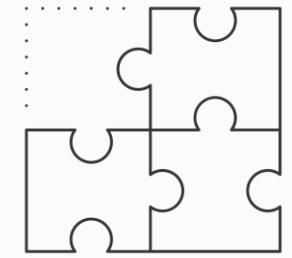
Q&A



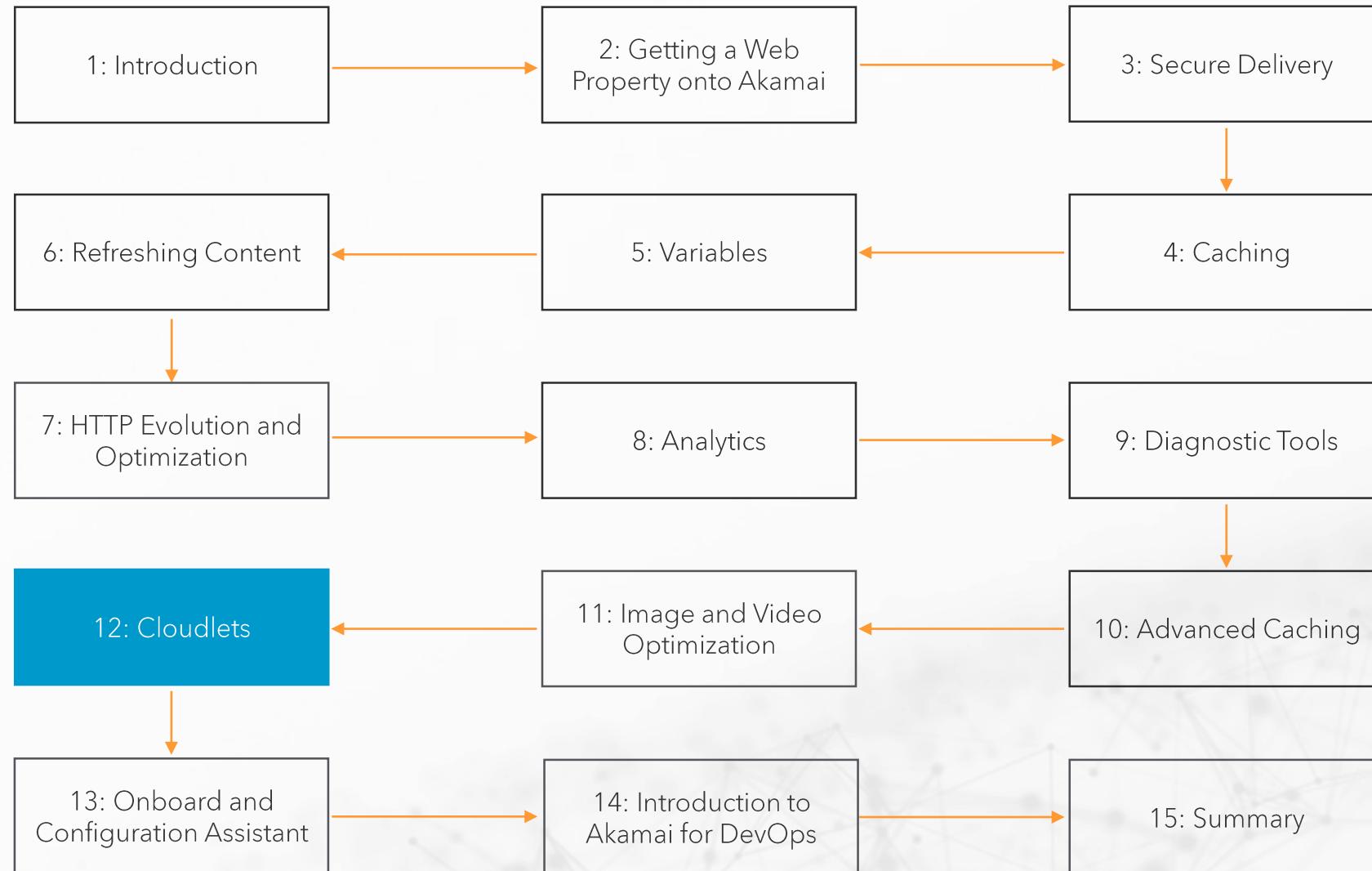
Summary

Image Optimization

- Challenges of images online
- Image Manager
 - IM Policy Manager
- Adaptive Image Compression



Agenda



MODULE 12

Cloudlets



Module Objectives

At the end of this module, you will understand:

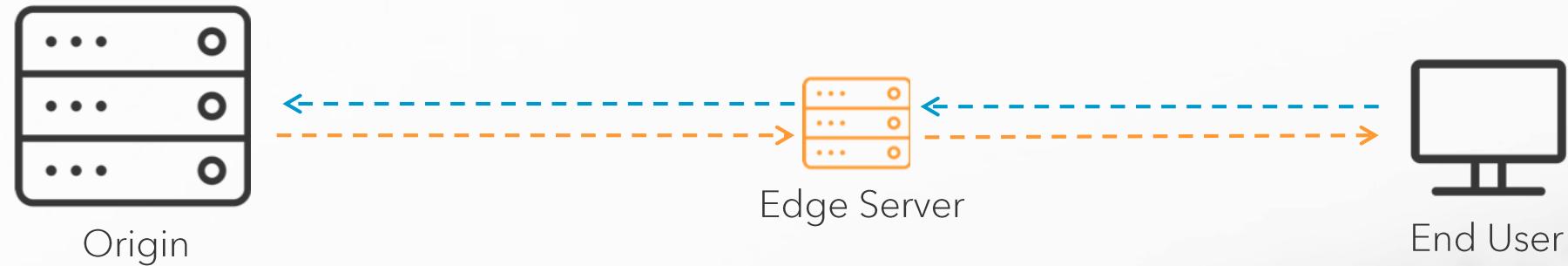
- The Cloudlets platform.
- Edge Redirector Cloudlet.
- Request Control Cloudlet.
- Application Load Balancer Cloudlet.



Cloudlets - Introduction

What is a Cloudlet?

A value-added application, designed to solve specific business challenges



Place your site's business logic at the "edge" of the Akamai Intelligent Platform.

Cloudlets Policy Manager

[Manage Cloudlets Policies](#)[Manage Load Balancing ...](#)[Reporting](#)[Console](#)

Cloudlets Policy Manager

[Help](#)

Create and manage the different Cloudlet policies configured for your account.

Cloudlets are small, value-added applications which solve specific business challenges.

[What's New in Cloudlets](#)

Updated: June 1, 2018

 Start typing to filter list[Color Key](#)[Create New Policy](#)

Cloudlets are **self-provisioned** and **self-configured** via the Cloudlets Policy Manager.

You can also manage Cloudlets via included **{OPEN}** APIs.

And **safely extend control** to other teams (IT, Marketing, Operations).

Cloudlet Configuration

When you configure a Cloudlet in the Cloudlets Policy Manager, you set up the following:

Step 1: Create cloudlet policy within Cloudlets Policy Manager

Step 2: Add cloudlet behavior within your Property Manager

Step 3: Activate your cloudlet within Cloudlets Policy Manager

Step 4: Activate your Property Manager configuration

Cloudlets Portfolio

Application Load Balancer is a dual-layer load balancer in the cloud that provides granular control and two types of failover logic.

Edge Redirector enables you to easily manage and offload high volumes of URL redirects.

Phased Release enables you to perform blue/green deployments or canary releases with built-in proactive failover to reduce risk during continuous software release activities.

Visitor Prioritization enables you to create a virtual HTML waiting room during or ahead of peak traffic events.

API Prioritization enables you to respond rapidly to changes in user demand and improve consistency when APIs are underperforming.

Audience Segmentation enables you to define various audiences for A/B testing, multivariate testing, and personalization.

Forward Rewrite helps improve SEO rankings by easily converting dynamic URLs into semantic URLs with a static-looking format.

Request Control enables you to offload unqualified traffic from your origin and provide conditional visitor access by controlling which requests your site responds to at the edge.

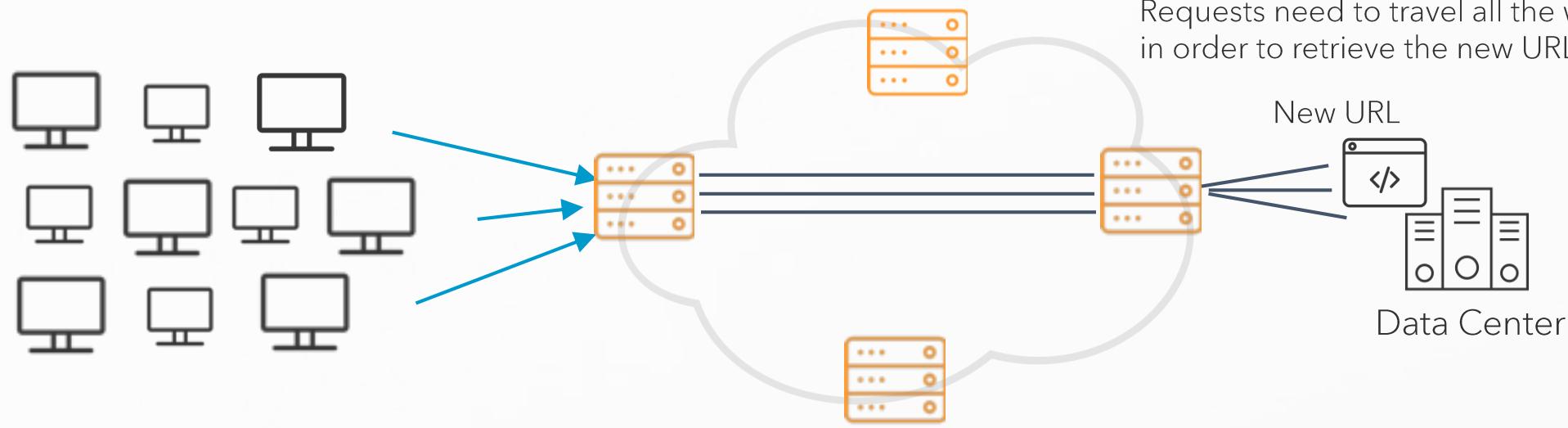
Edge Redirector Cloudlet

The Edge Redirector Cloudlet enables you to easily manage and offload high volumes of URL redirects. Redirects are generated at the edge without the need to build origin infrastructure to return a response, which allows for faster responses and scalability. Additionally, Edge Redirector provides a tool to find and remove unused redirect rules. You can define redirects using the Cloudlet Policy Manager UI; by importing a CSV file containing the source, target, and type of redirect; or using the Cloudlet API.

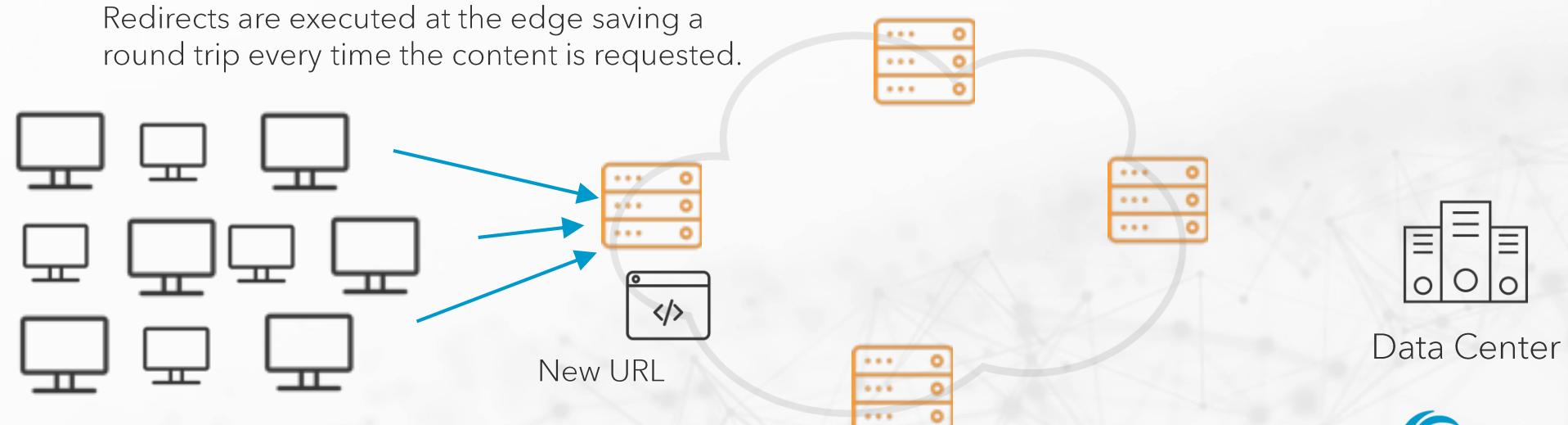
Use-Cases

- Redirecting URLs while preserving SEO
- URL redirects as part of regular web operations
- Extending URL redirect control to marketing teams to reduce IT ticket requests
- Creating vanity URLs
- Managing marketing campaigns

How Edge Redirector Works



Redirects are executed at the edge saving a round trip every time the content is requested.



Edge Redirector Configuration

Step 1: Create cloudlet policy within Cloudlets Policy Manager

- Choose cloudlet type
- Name it (best to choose naming convention)
- Create access group association
- Define “rules” within your cloudlet
 - First rule wins rule applies
- Cloudlet policy is versioned

Create Policy

| | |
|-------------------|------------------------|
| Cloudlet Type | Edge Redirector |
| Policy Name | Test_Redirector_Policy |
| Notes | Example policy |
| Group Association | QA |

Cancel **Create Policy**

Edge Redirector Configuration

Step 2: Add cloudlet behavior within your Property Manager

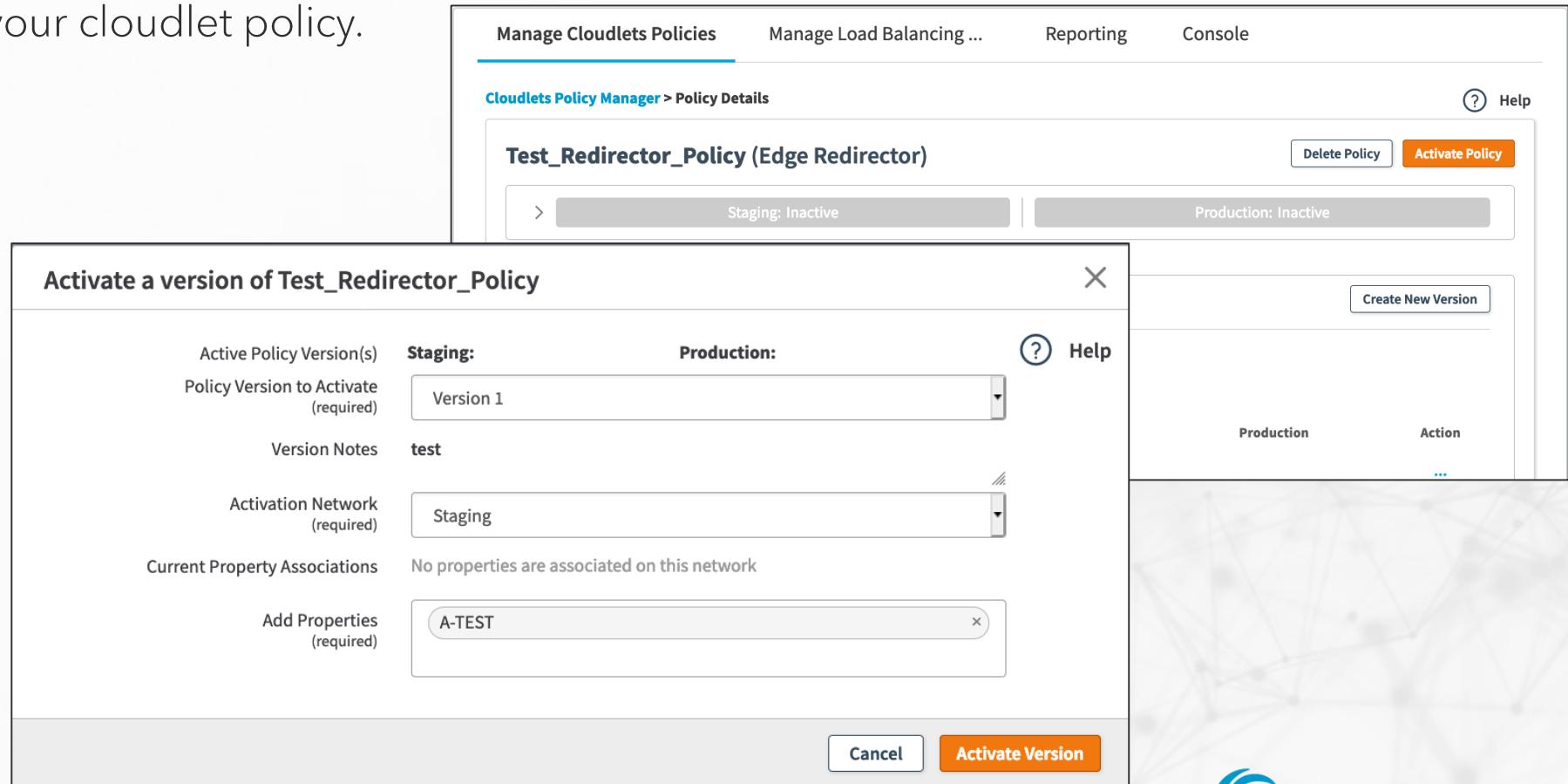
- Add the cloudlet behavior you want within your PM configuration.
- Reference the name of the cloudlet policy you created in Step.



Edge Redirector Configuration

Step 3: Activate your cloudlet within Cloudlets Policy Manager

- You can activate your cloudlet policy.



The screenshot shows the Akamai Cloudlets Policy Manager interface. At the top, there are tabs for 'Manage Cloudlets Policies', 'Manage Load Balancing ...', 'Reporting', and 'Console'. Below the tabs, a breadcrumb navigation shows 'Cloudlets Policy Manager > Policy Details'. The main title is 'Test_Redirector_Policy (Edge Redirector)'. On the right, there are buttons for 'Delete Policy' and 'Activate Policy'. Below the title, there are two status indicators: 'Staging: Inactive' and 'Production: Inactive'. A large modal window is open in the foreground, titled 'Activate a version of Test_Redirector_Policy'. The modal has several input fields:

- 'Active Policy Version(s)': A dropdown menu showing 'Version 1'.
- 'Policy Version to Activate (required)': A dropdown menu showing 'test'.
- 'Version Notes': A text input field containing 'test'.
- 'Activation Network (required)': A dropdown menu showing 'Staging'.
- 'Current Property Associations': A text input field showing 'No properties are associated on this network'.
- 'Add Properties (required)': A text input field containing 'A-TEST'.

 At the bottom of the modal are 'Cancel' and 'Activate Version' buttons.

Edge Redirector Configuration

Step 4: Activate Property Manager configuration.

- You can activate your configuration after the cloudlet become active on the network.

Edge Redirector - Simple Rule

Rule List ⓘ Policy Analysis

Start typing to filter list

Upload Rules Download Rules Update Order Add Rule Save Changes

| <input type="checkbox"/> Order ↑ | Na... | Match | Redirect URL | Stat... | Copy to Re... | Start | End |
|-----------------------------------|-------|-------|--------------|---------|---------------|-------|-----|
| ⓘ No data available | | | | | | | |

Create a Rule ⓘ Help X

Rule Name: Simple redirect

Enabled:

Always On:

Source URL (required):

Redirect URL (required):

Redirect Type (required):

Copy Query String:

Relative Redirect Options:

Show Advanced View Cancel Save Rule

Edge Redirector - Advanced Rule

Protocol

AND

Hostname

demo.akau.akamaiuweb.com

AND

URL Path

/

Case Sensitive
 Allow Wildcards * (Zero Or More) And ? (Any Character)

Create a Rule Help X

GENERAL

Rule Name: Simple redirect

Enabled:

Always On:

IF

Protocol

AND

Hostname

demo.akau.akamaiuweb.com

AND

URL Path

/

Case Sensitive
 Allow Wildcards * (Zero Or More) And ? (Any Character)

THEN

Redirect URL (required):

Redirect Type (required):

Copy Query String:

Relative Redirect Options:

Edge Redirector Rules - Advanced Regex Rule

URL Regular Expression ▾

Regular Expression (Regex):
(http|https://www.akamaiuweb.com/)

Redirect URL:
\1://www.akamai.com/university

Hide Regex Tester (?

Test URL:
http://www.akamaiuweb.com/

Captured groups generated by regex:

| ID | Captured Substring |
|----|--------------------|
| \1 | http |

Resulting URL based on substitution pattern:
http://www.akamai.com/university

Validate

Regular expression

Test URL

Testing results

Edge Redirector Testing - Policy Cost Analysis

Policy Analysis will:

- Validate your redirect rules
- Computes the “cost” (CPU/computational resources)
- Rules with cost of more than 1,000,000 not work

Rule List **Policy Analysis**

The analysis uses the hostname currently defined in this policy version. You can also select a different hostname or Akamai server IP address for analysis.

Rules highlighted in red either are too complex, or have mismatched rule IDs, or do not have a redirect URL.

Select to specify a hostname to test, an Akamai server IP address on which to test, or both.

demo.akau.akamaiuweb.com 23.50.55.24 [?](#)

Rerun Analysis

Start typing to filter list [Download Report](#) [Show Only Errors](#) [Show All Rules](#) [Hide Selected Rules](#)

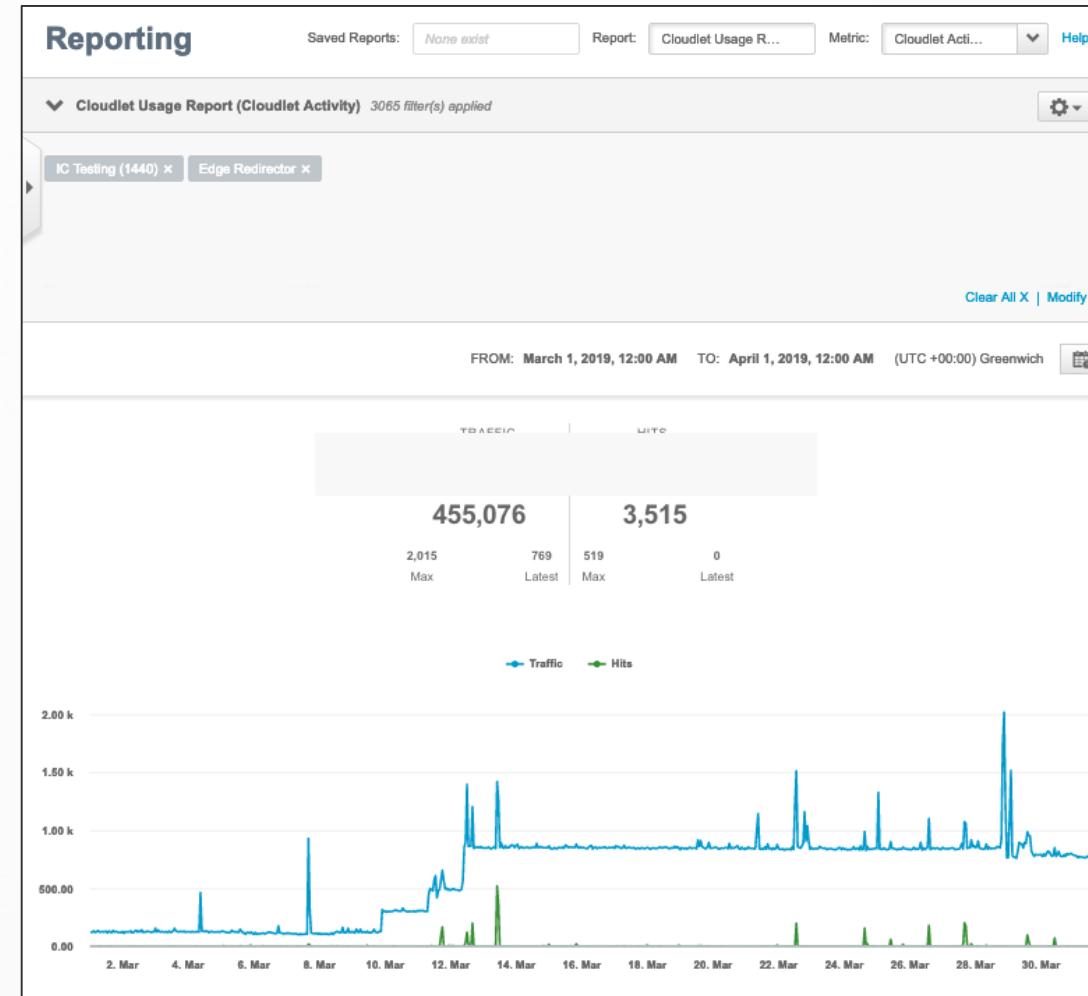
| <input type="checkbox"/> Or... | Redirect URL | Status | Executed? | Cost ↓ |
|--------------------------------|--|--------|-----------|--------|
| <input type="checkbox"/> | ▶ 2 http://www.akamai.com/akamai_university.html | 302 | Yes | 134844 |
| <input type="checkbox"/> | ▶ 1 | 200 | No | 91358 |

Edge Redirector Testing

- Once everything is deployed to one of the networks you can start testing.
- Verify that the right location and response code is served.
- Looking at Server header you can identify source of the response.

```
curl http://demo.akau.akamaiuweb.com/
* Connected to demo.akau.akamaiuweb.com
(23.50.55.24) port 80 (#0)
> GET / HTTP/1.1
> Host: demo.akau.akamaiuweb.com
>
< HTTP/1.1 302 Moved Temporarily
< Server: AkamaiGHost
< Content-Length: 0
< Location:
http://www.akamai.com/akamai_university.htm
1
< Date: Thu, 11 Apr 2019 13:53:00 GMT
```

Cloudlets Usage Reports - Edge Redirector



Why Edge Redirector Cloudlet?

You can configure redirects in Property Manager, so why use a dedicated cloudlet for this?



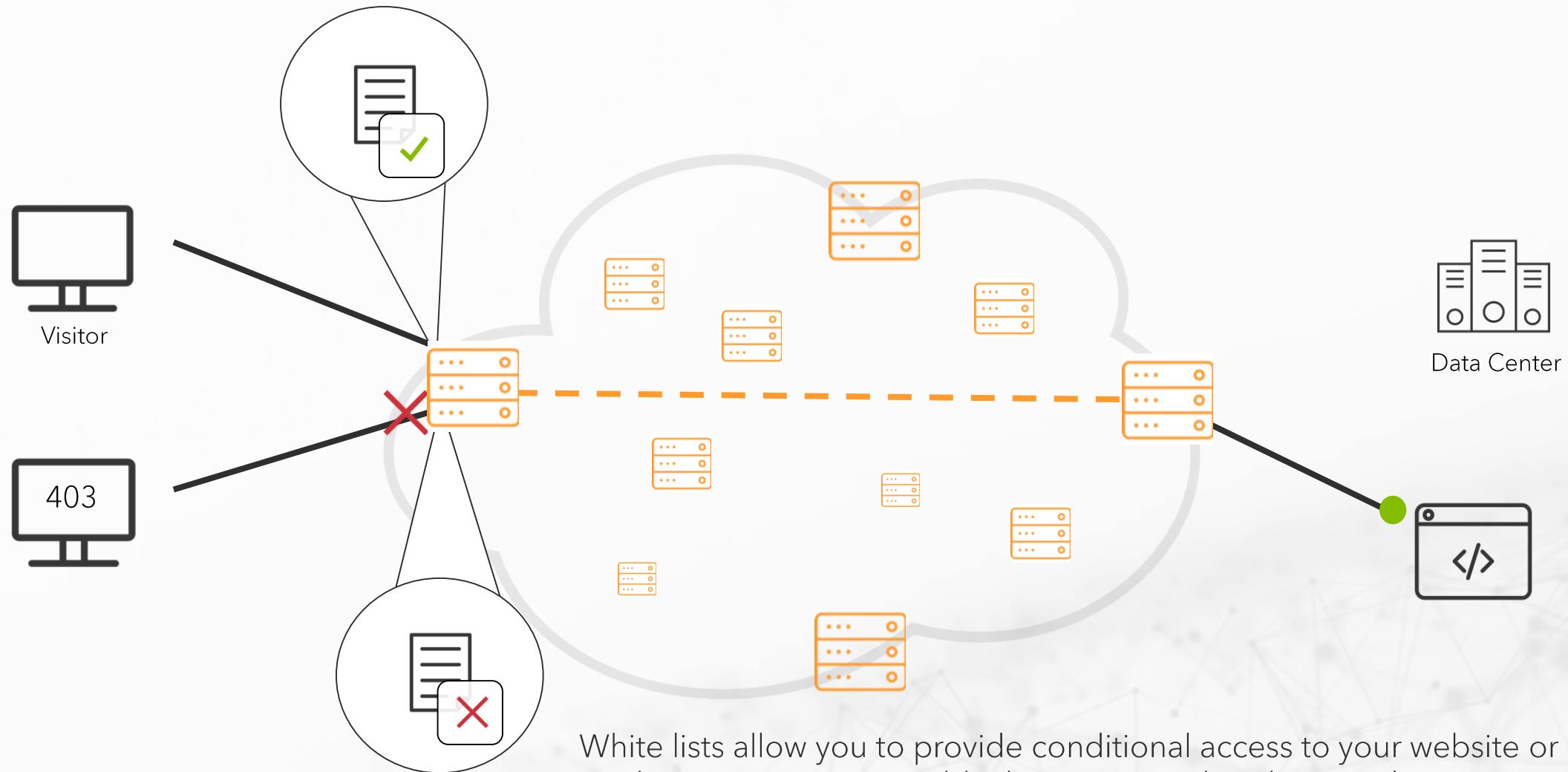
Request Control Cloudlet

The Request Control Cloudlet enables you to offload unqualified traffic from your origin and provide conditional visitor access by controlling which requests your site responds to at the edge. By configuring this Cloudlet to detect variables like geolocation, IP address, header value, and request type, unwanted traffic will be denied access to your website or application.

Use Cases:

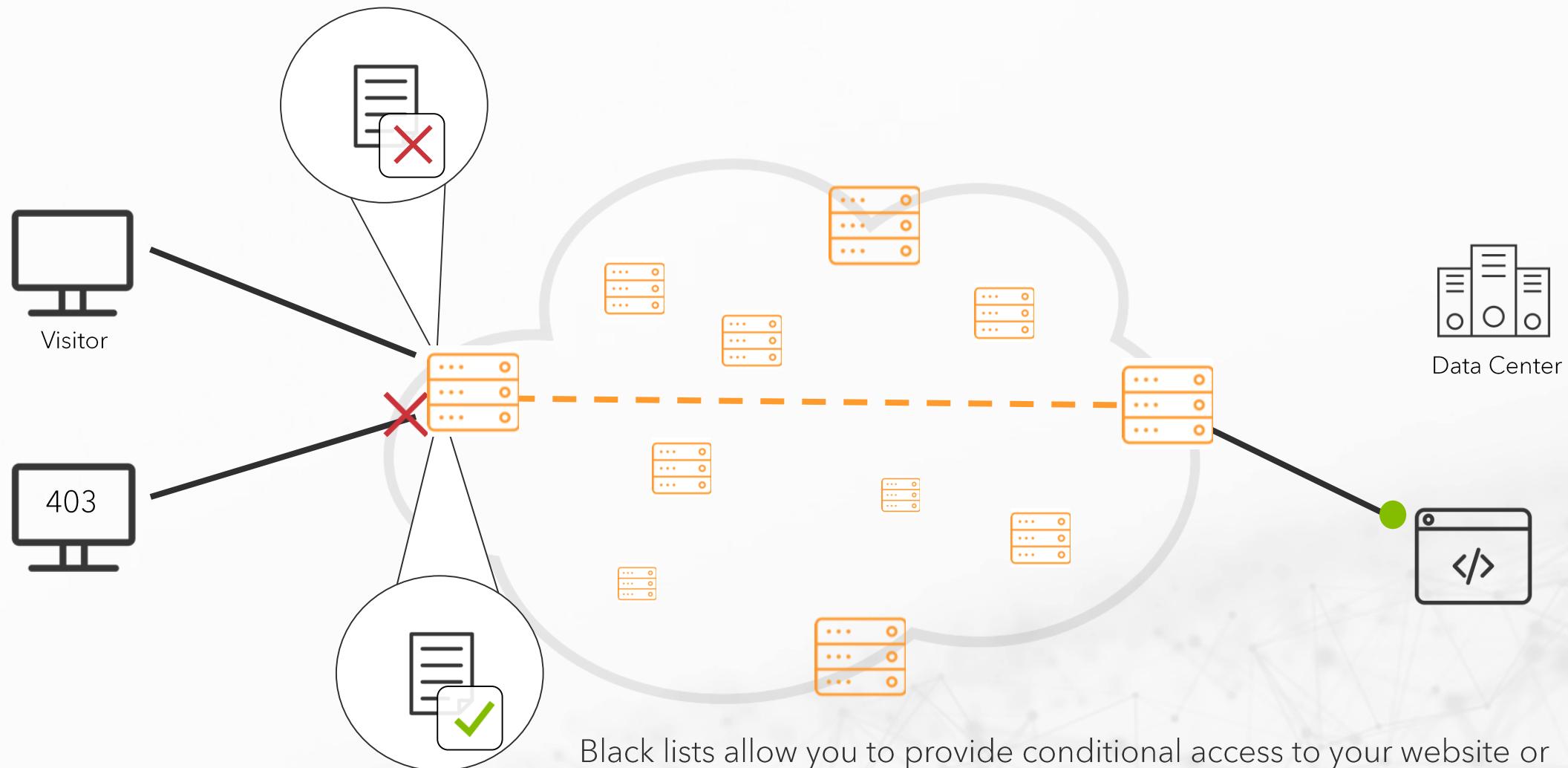
- E-commerce sites with regional online sales
- Website maintenance, QA testing, restricted information
- Blocking users who post unwanted comments
- Denying traffic from low-priority geographies to free up capacity for the intended audience
- Denying requests from a geographic region that has a high probability of illegitimate traffic and a low priority to your business

How Request Control Works (White List)



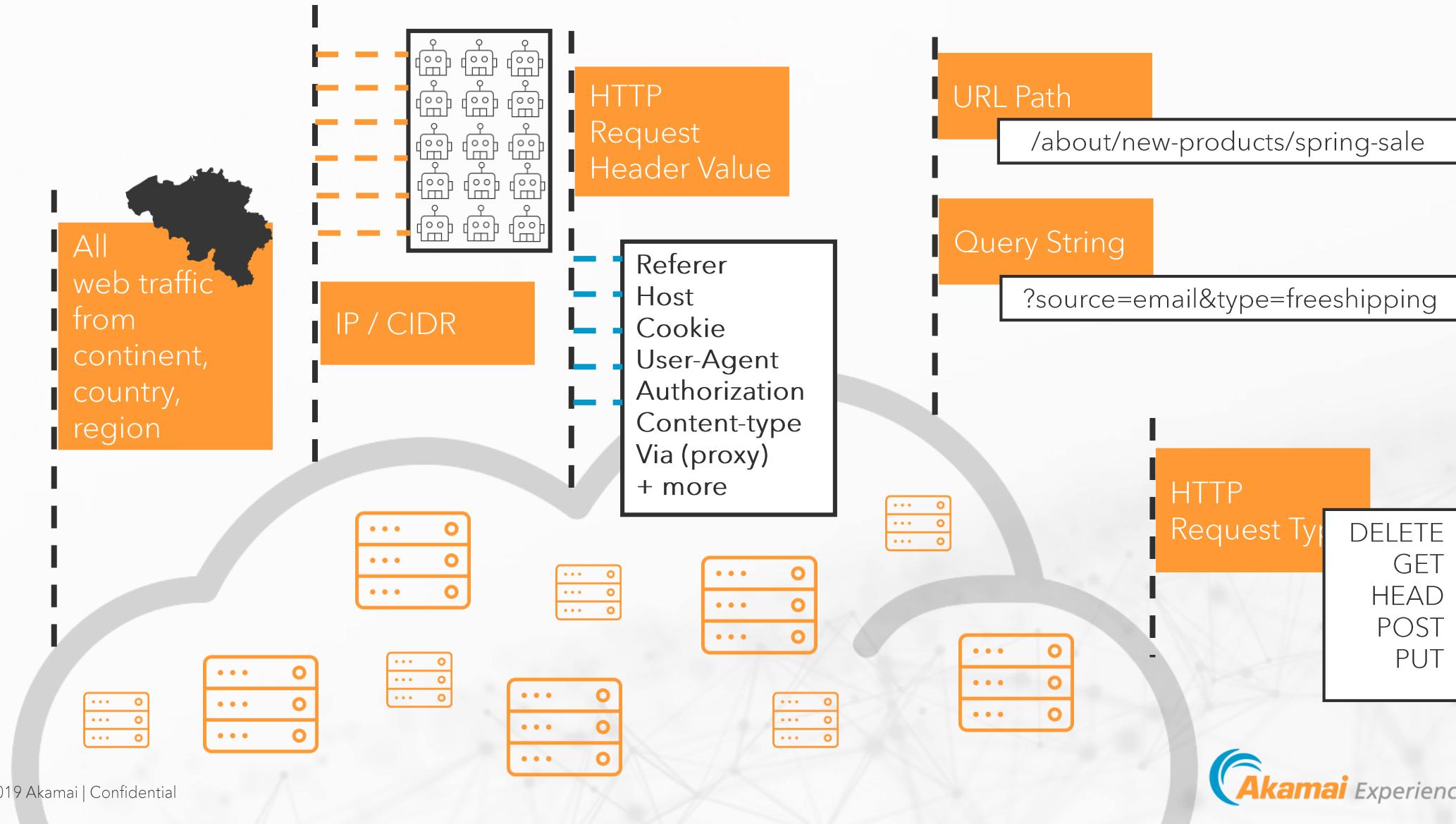
White lists allow you to provide conditional access to your website or application permitting or blocking users at the Akamai Edge.

How Request Control Works (Black List)



Black lists allow you to provide conditional access to your website or application permitting or blocking users at the Akamai Edge.

Design Policies with Flexible Match Conditions



Request Control Configuration - Policy Rule

IF

IP/CIDR (?) +

is one of

Enter IP addresses and CIDR blocks (?)

Clear

and the client IP address

is the Connecting IP Address

THEN

Allow / Deny*: Branded Response

Cancel Save Rule

Request Control Configuration - Property Behavior

Request Control Cloudlet

Enable **On**

Policy Name **Test_REQUEST_Control** 

Cloudlet policy name

Enable Branded Response Page **On**

Status Code **503 Service Unavailable**

NetStorage **Test NetStorage - (testns2.download.akamai.com/175844)** 

Branded Response Path and File Name **/brand/response.html**

Branded Response Cache TTL (min) **5** 

Location and details of fixed branded file

Request Control Cloudlet Use

When might you use the Request Control Cloudlet?



Cloudlets Origins

Application Load Balancer, Audience Segmentation and Phased Release Cloudlets use a feature called Cloudlets Origins.

Using [Cloudlets Origins](#), you can define alternate origin servers and control which origin to use from within your Application Load Balancer Cloudlet.

To configure this, add the Cloudlets Origin Group rule template from the list of rule templates.

Add new Rule

Available Rules (by Category)

Default Templates ▾

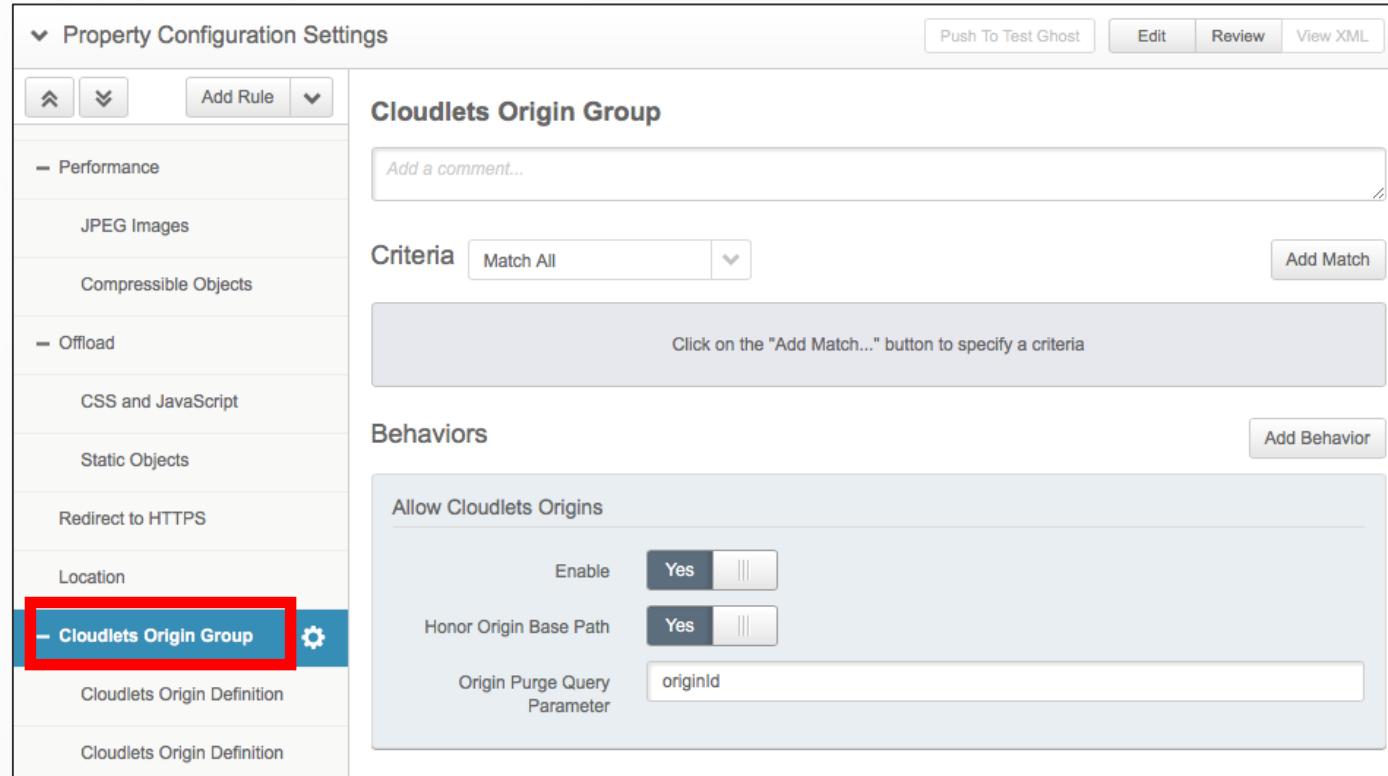
Cloudlets Origin

Cloudlets Origin Definition

Cloudlets Origin Group

Configuring Cloudlets Origins

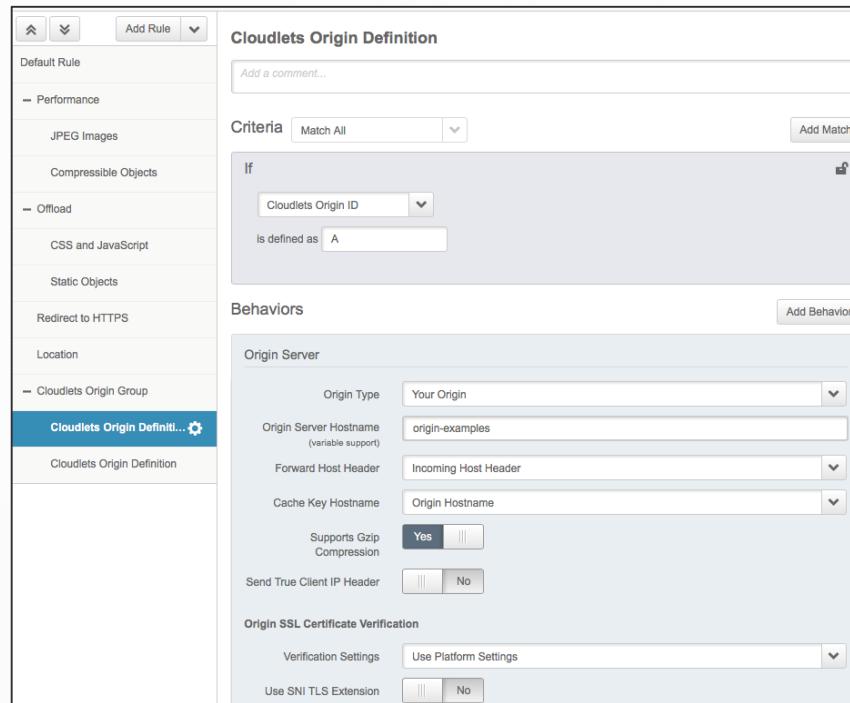
Once adding the Cloudlets Origin Group rule type you have the following structure. No configuration is required on the main rule:



The screenshot shows the 'Property Configuration Settings' interface for a Cloudlets Origin Group rule. The left sidebar lists various rule types, with 'Cloudlets Origin Group' highlighted and enclosed in a red box. The main panel is titled 'Cloudlets Origin Group' and contains sections for 'Criteria' (set to 'Match All') and 'Behaviors'. The 'Behaviors' section is titled 'Allow Cloudlets Origins' and includes settings for 'Enable' (set to 'Yes'), 'Honor Origin Base Path' (set to 'Yes'), and 'Origin Purge Query Parameter' (set to 'originId').

Configuring Cloudlets Origins

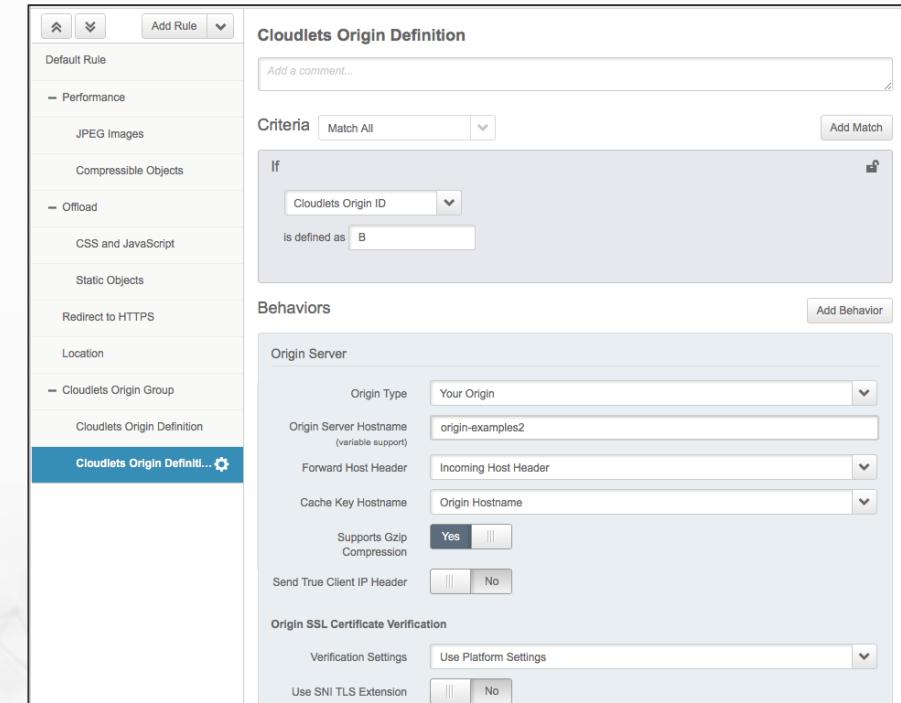
Each Cloudlet Origin Definition sub-rule contains a Cloudlets Origin ID, which you define and will later use within your Cloudlets, as well as an Origin Server definition, specific to that specific Origin ID.



The screenshot shows the 'Cloudlets Origin Definition' configuration page. The 'Criteria' dropdown is set to 'Match All'. The 'Behaviors' section contains an 'Origin Server' configuration with the following fields:

- Origin Type: Your Origin
- Origin Server Hostname: origin-examples
- Forward Host Header: Incoming Host Header
- Cache Key Hostname: Origin Hostname
- Supports Gzip Compression: Yes
- Send True Client IP Header: No

At the bottom, the 'Origin SSL Certificate Verification' section includes 'Verification Settings' (Use Platform Settings) and 'Use SNI TLS Extension' (No).



The screenshot shows the 'Cloudlets Origin Definition' configuration page. The 'Criteria' dropdown is set to 'Match All'. The 'Behaviors' section contains an 'Origin Server' configuration with the following fields:

- Origin Type: Your Origin
- Origin Server Hostname: origin-examples2
- Forward Host Header: Incoming Host Header
- Cache Key Hostname: Origin Hostname
- Supports Gzip Compression: Yes
- Send True Client IP Header: No

At the bottom, the 'Origin SSL Certificate Verification' section includes 'Verification Settings' (Use Platform Settings) and 'Use SNI TLS Extension' (No).

Application Load Balancer (ALB) Cloudlet

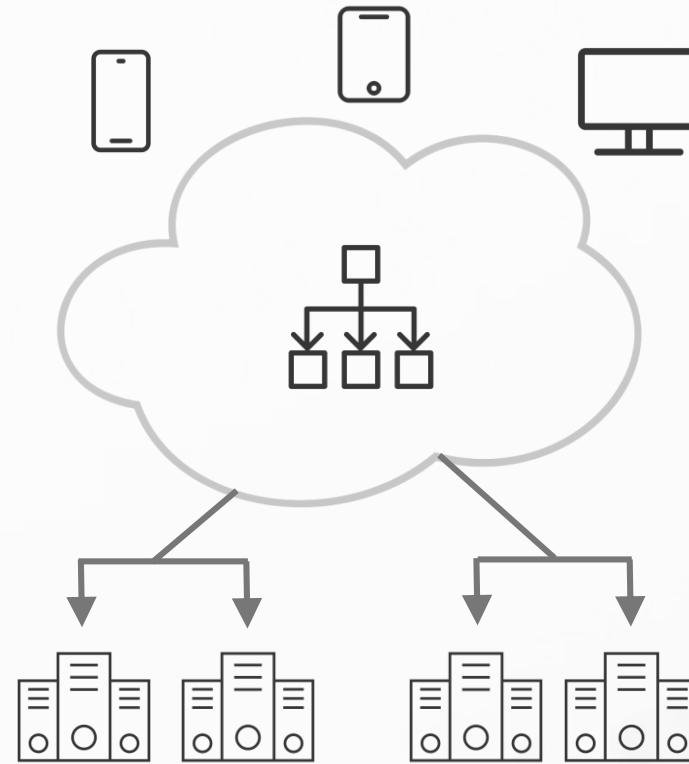
The Application Load Balancer Cloudlet is a dual-layer load balancer in the cloud that provides granular control and two types of failover logic. This gives you the option to split traffic at either the HTTP/S layer or the DNS layer. Using liveness tests, Application Load Balancer determines the health of your origin, and when requests return with configurable status codes, instant-failover is triggered and your traffic is immediately redirected to an available origin. In the case of all origins being down, Application Load Balancer provides a fully configurable down page.

Use Cases:

- Cloud migration
- Multi-cloud architectures
- Failover and load balancing between cloud regions
- E-commerce websites running from multiple origins
- Preparing for holiday traffic or sales
- Live streaming events

Application Load Balancer

A multi-layer load balancer leveraging attributes from both the application layer (Layer 7) and the DNS layer (Layer 3) providing control over traffic directed to various data sources.



Session Stickiness



Cookie Based Session Stickiness maintains control for consistent session stickiness behaviors

Origin Health Checks



Origin Health Checks w/ Seamless Failover and Outage Response

No Hardware, No Code



Reduce complexity & offload routing logic from your data tier to Akamai



Fast, Intelligent Routing

Akamai's SureRoute algorithm provides high availability and instant scale through dynamic route optimization and failover logic



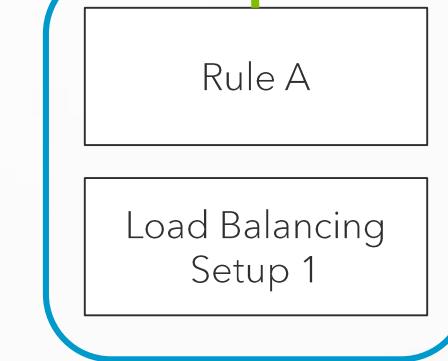
Easy to Setup

Easy to use interface, and **included API**, allow you to click to design and activate policies and/or extend cloudlet controls to operations staff

Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover

Simple Application Load Balancer
Setup -
one rule and one load balancing
setup

Application Load Balancer

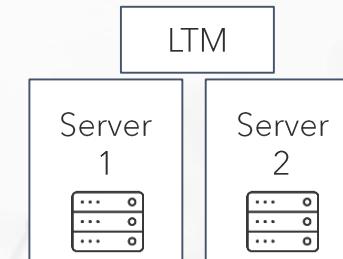


HTTP Layer Based Rules:

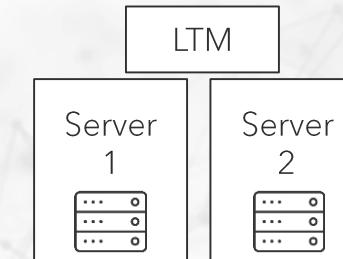
- User location data
- URL path
 - Ex. Micro services URL routing
- Cookie
- Device characteristics
 - Ex. Mobile users
- File Extensions
- Etc.

LTM → Local Traffic Manager (ex.
F5)

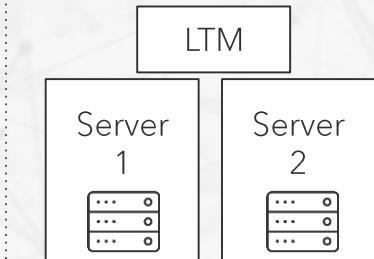
Data Center UK



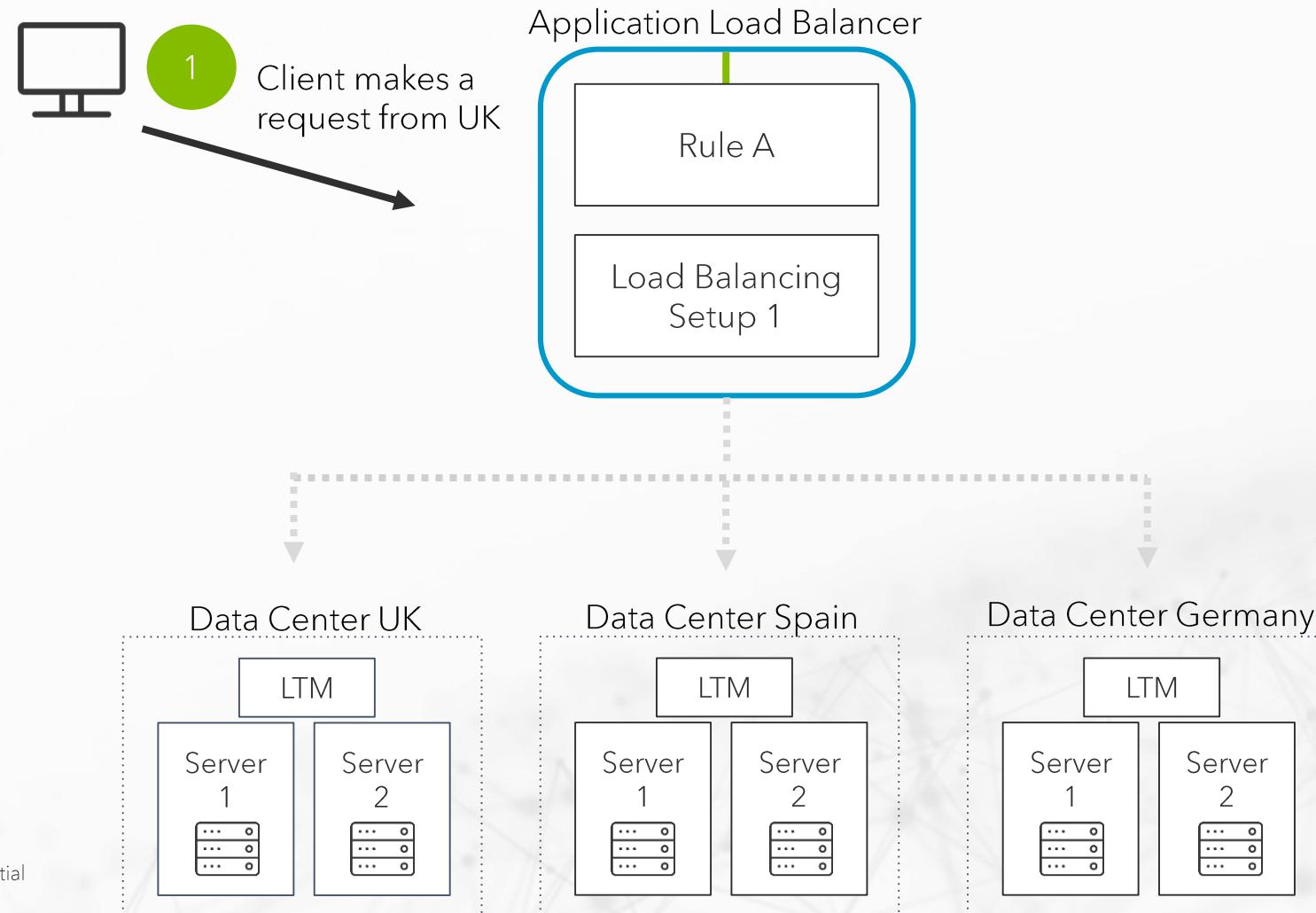
Data Center Spain



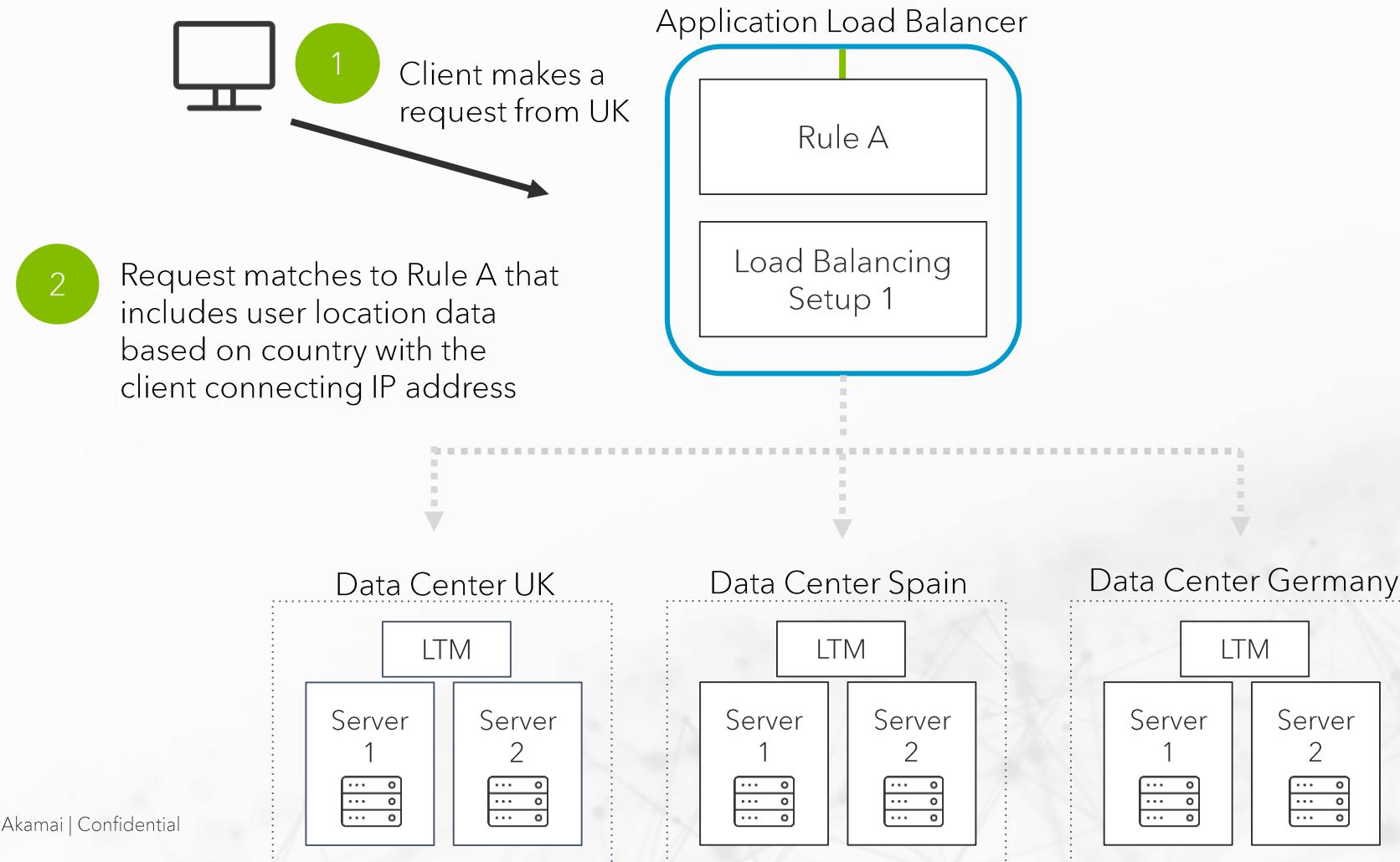
Data Center Germany



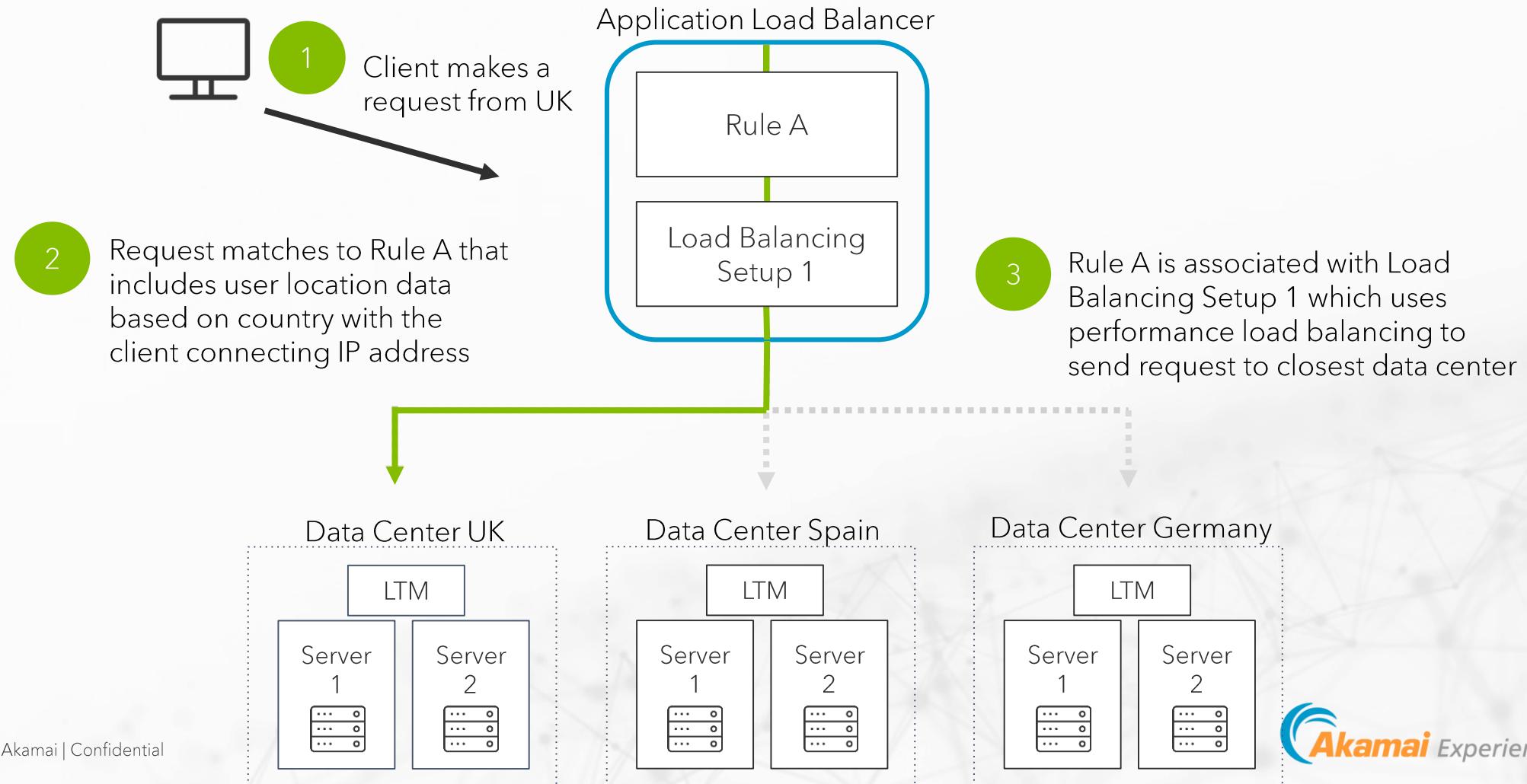
Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



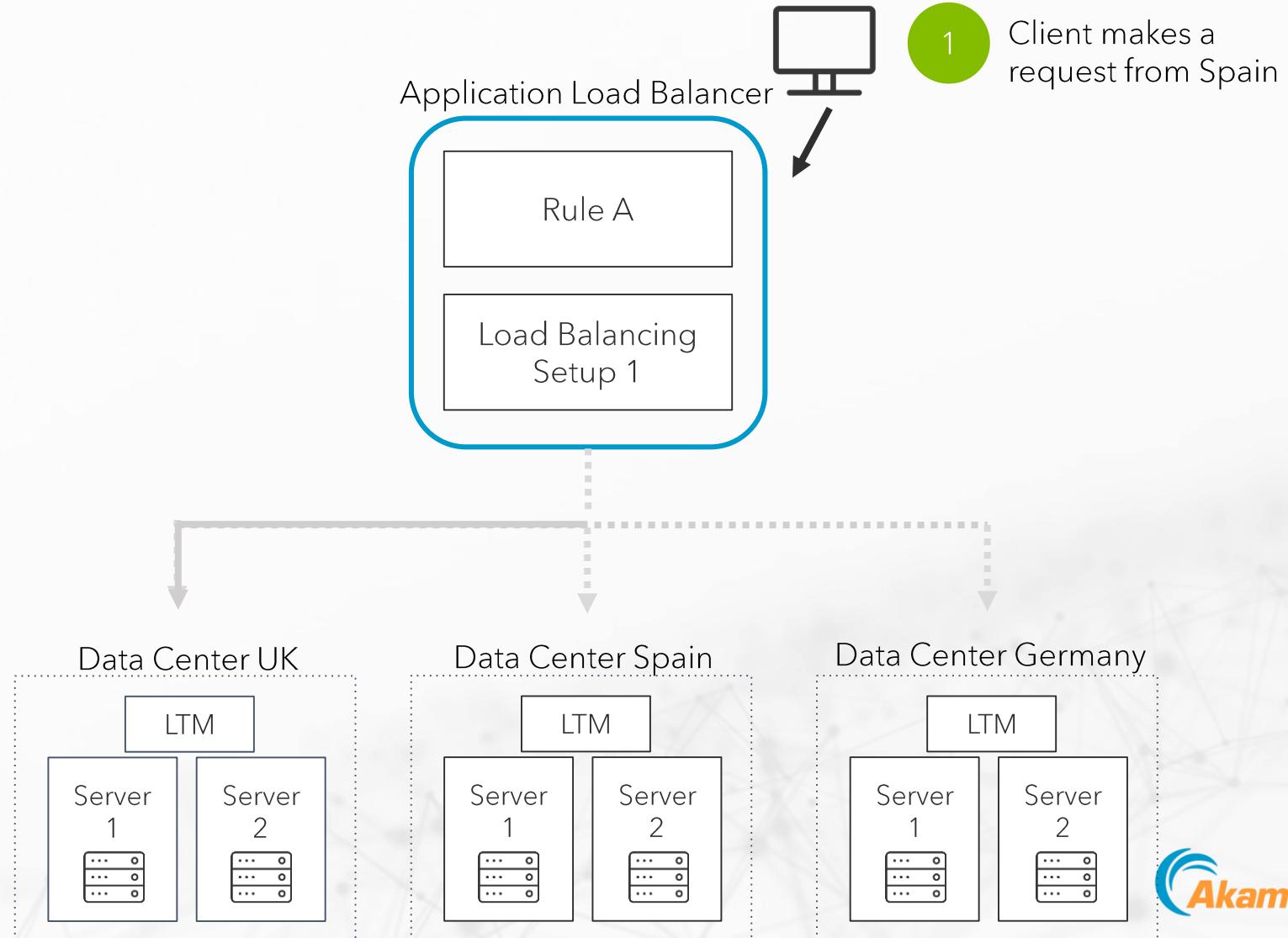
Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



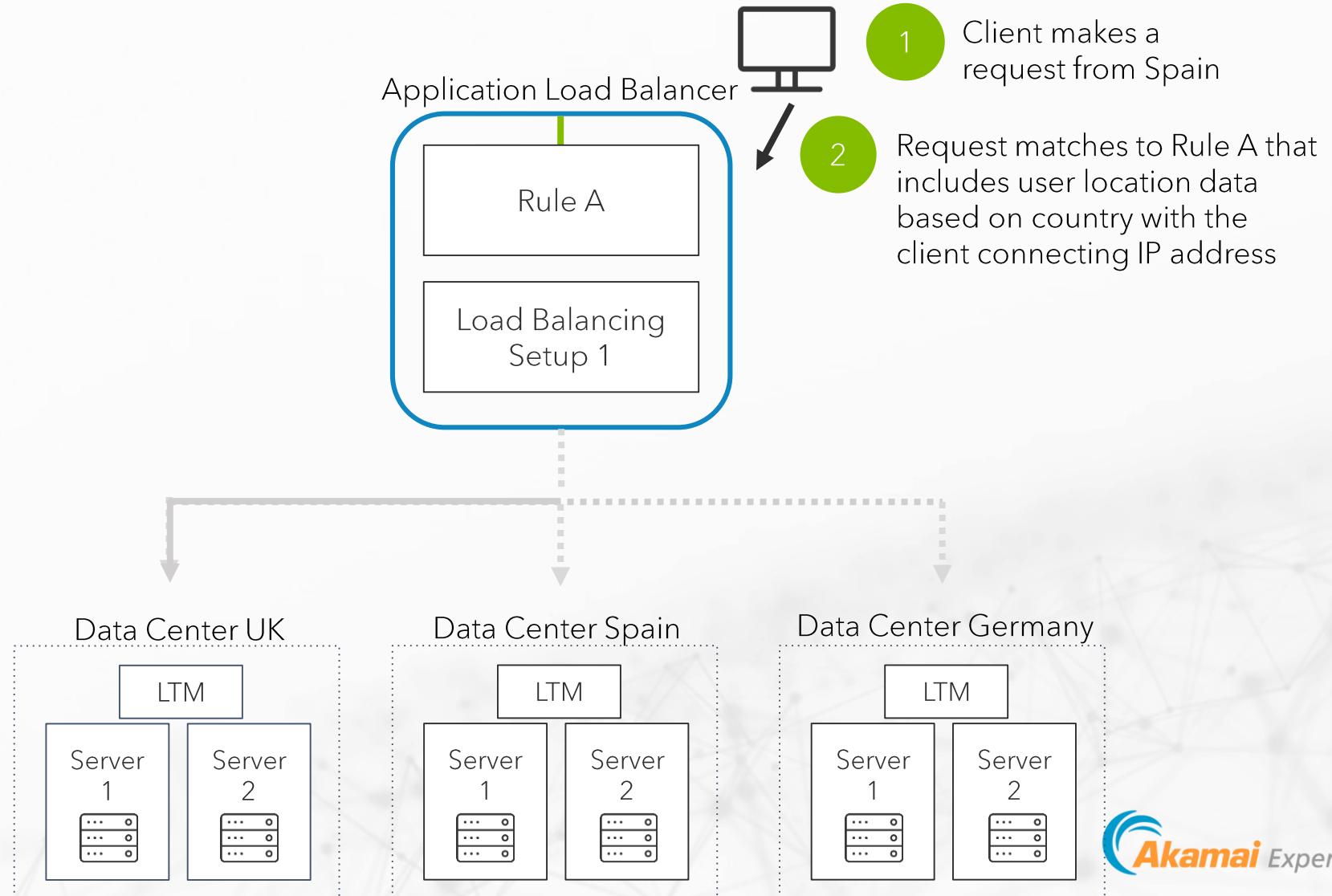
Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



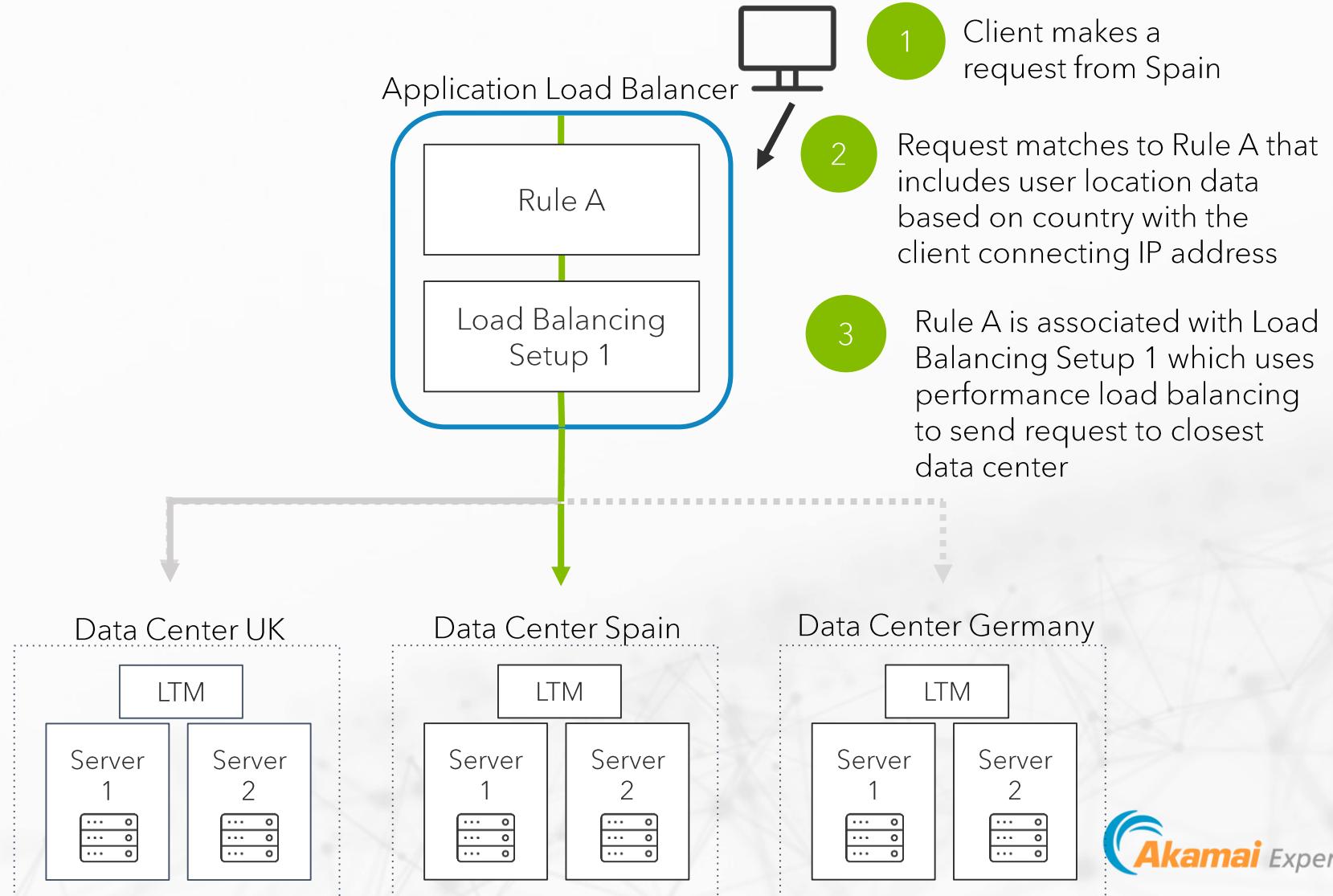
Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



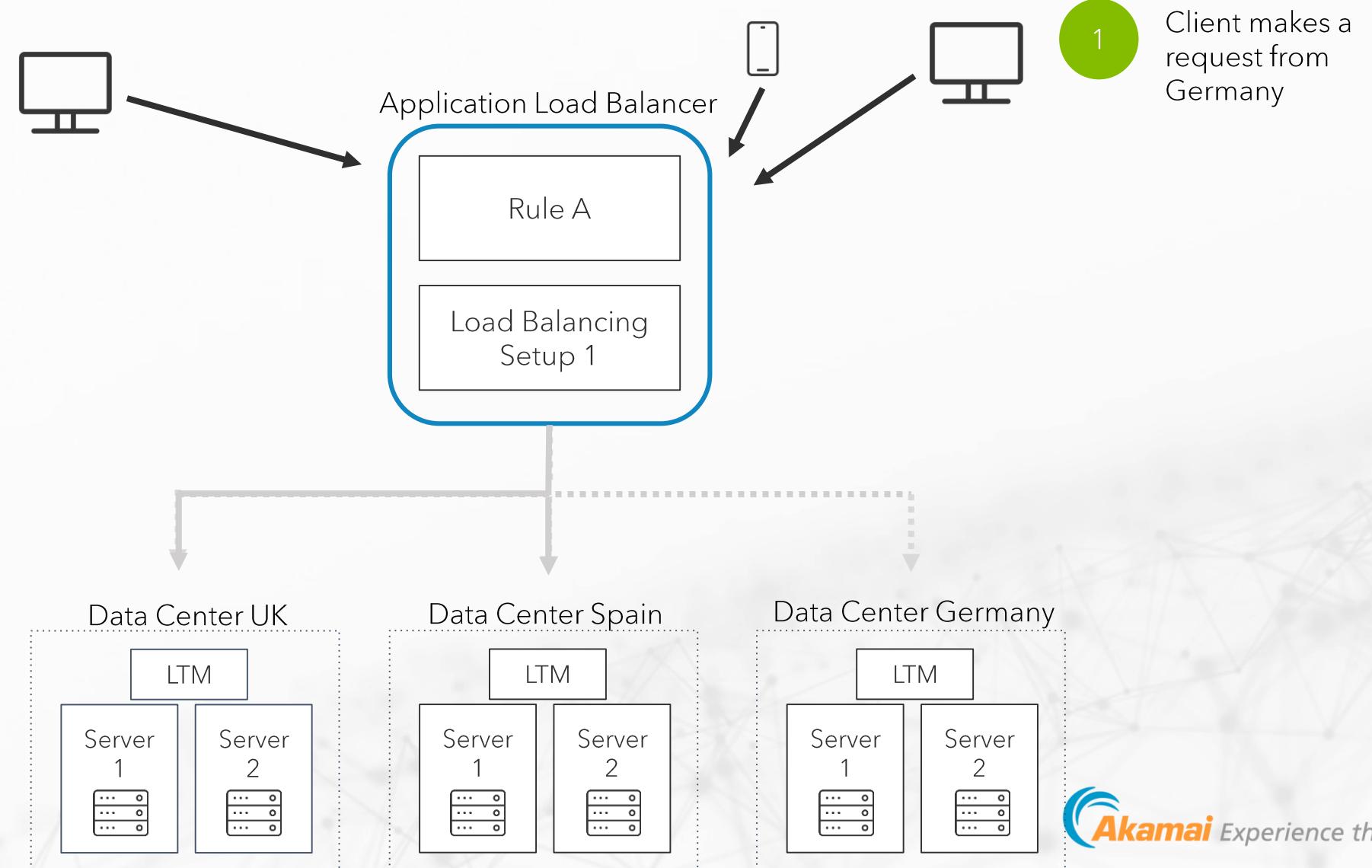
Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



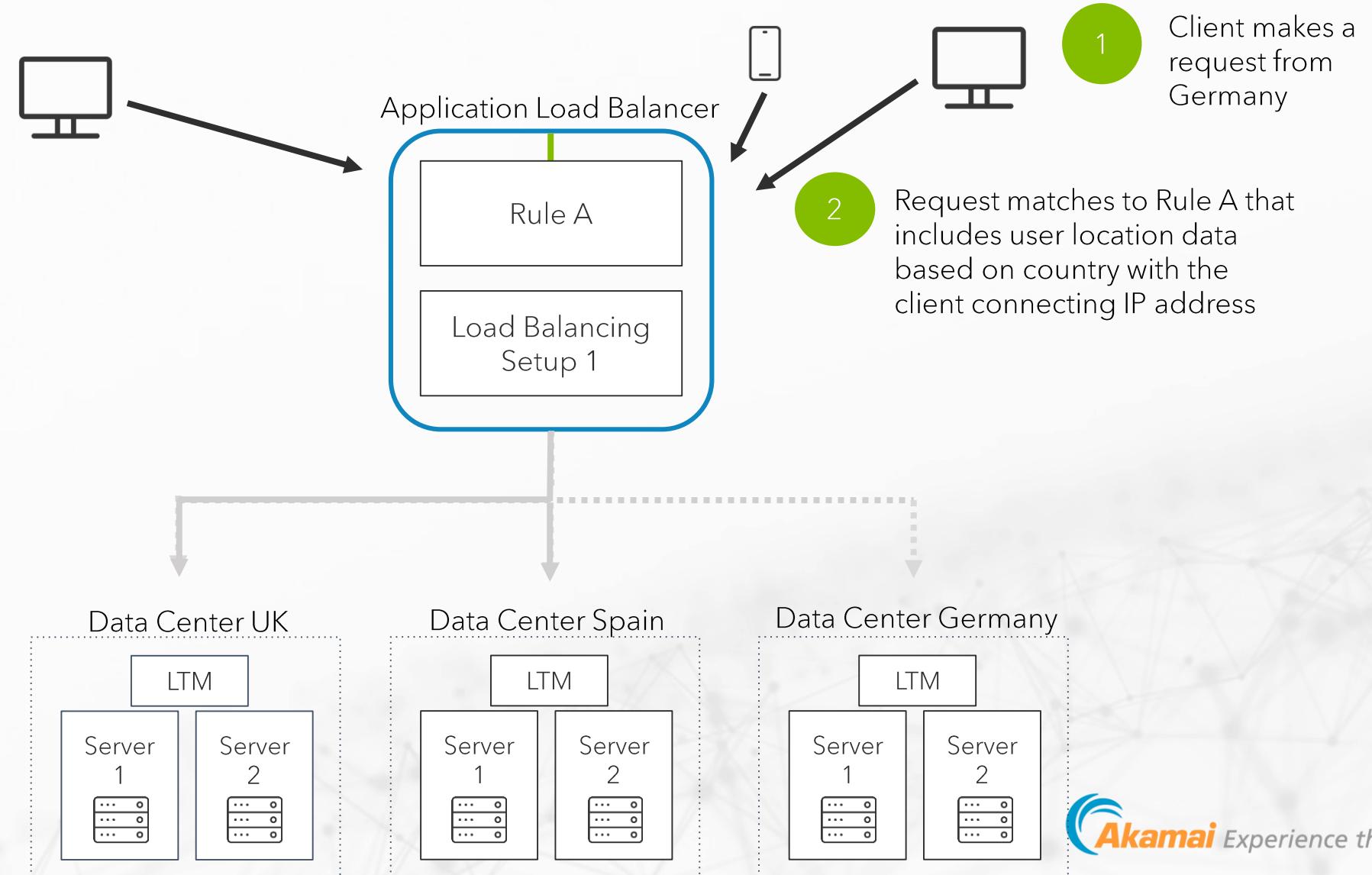
Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



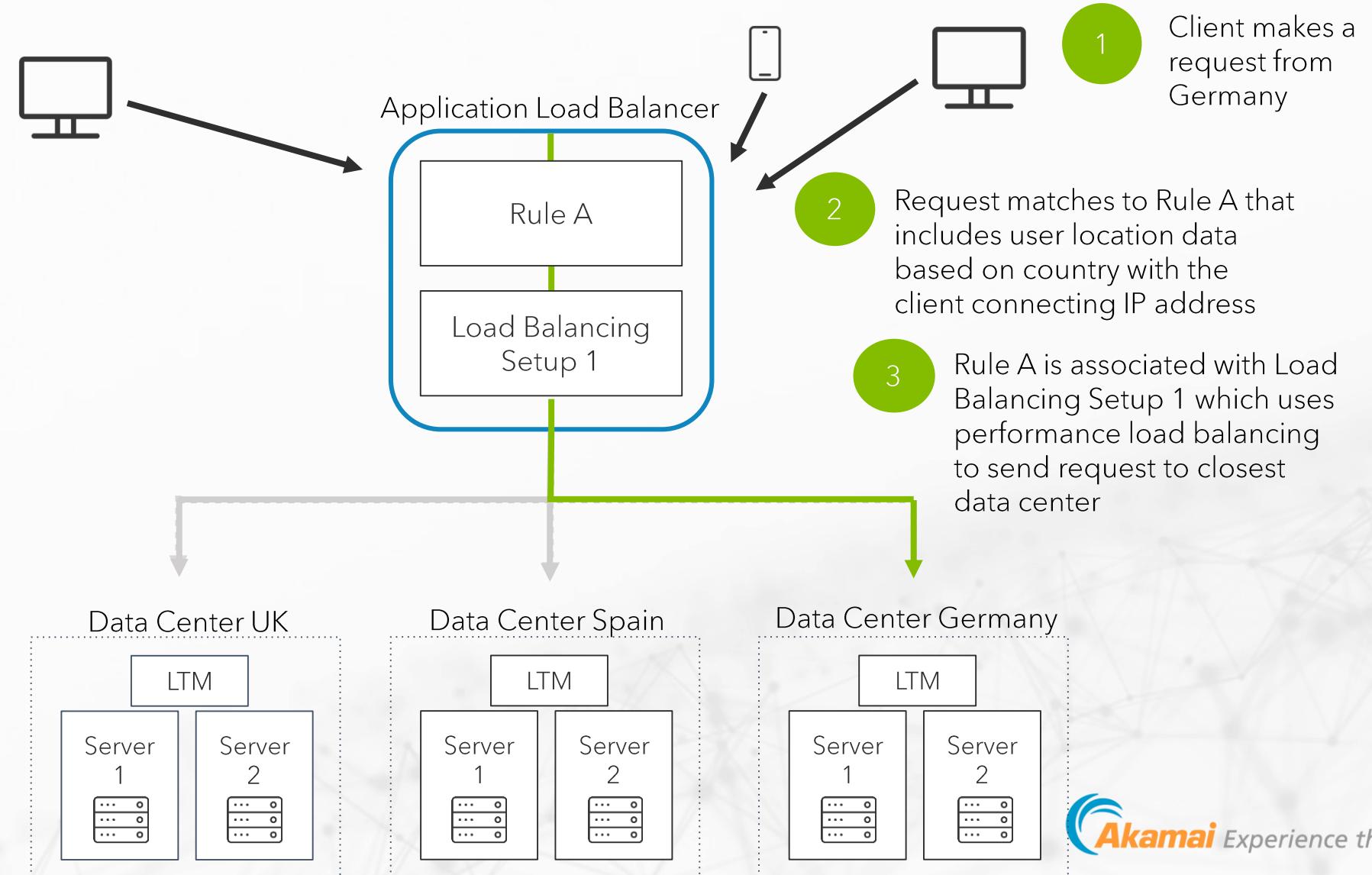
Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



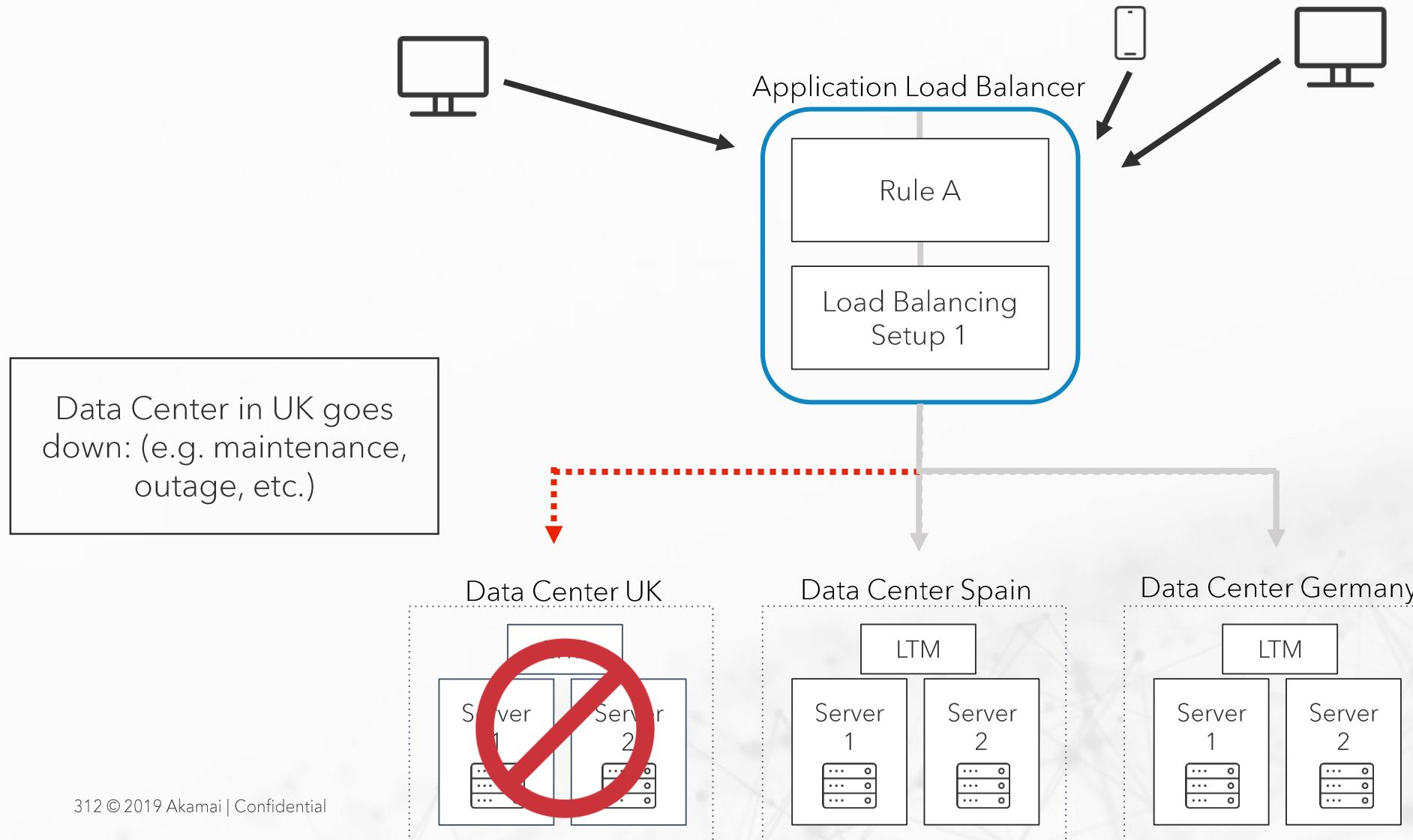
Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



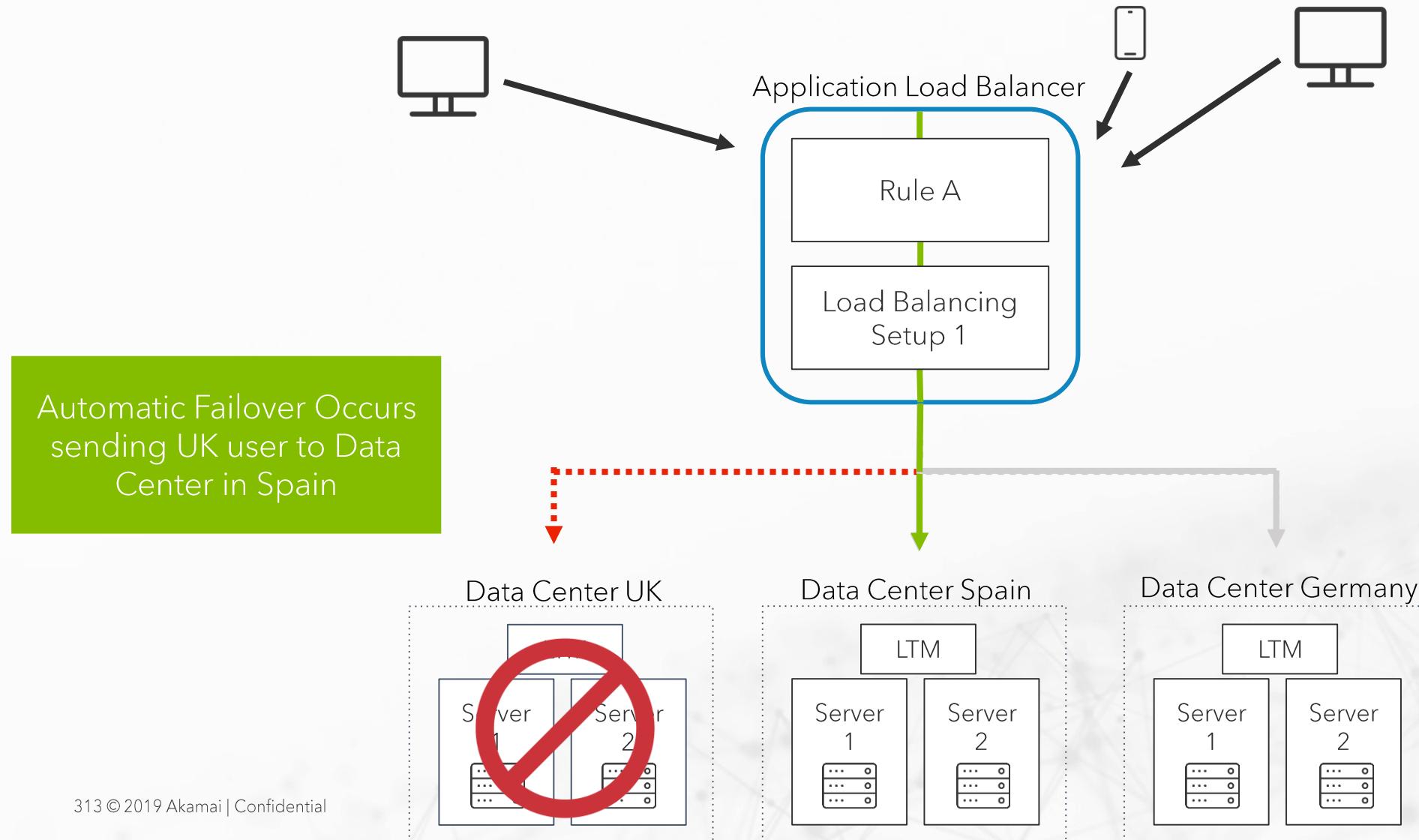
Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



Application Load Balancer Cloudlet Use Case: Multiple Data Centers + Session Stickiness + Failover



LAB 7

Cloudlets



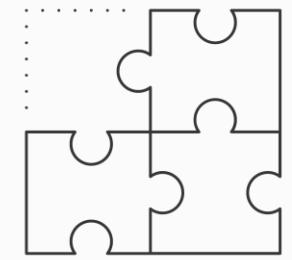
Q&A



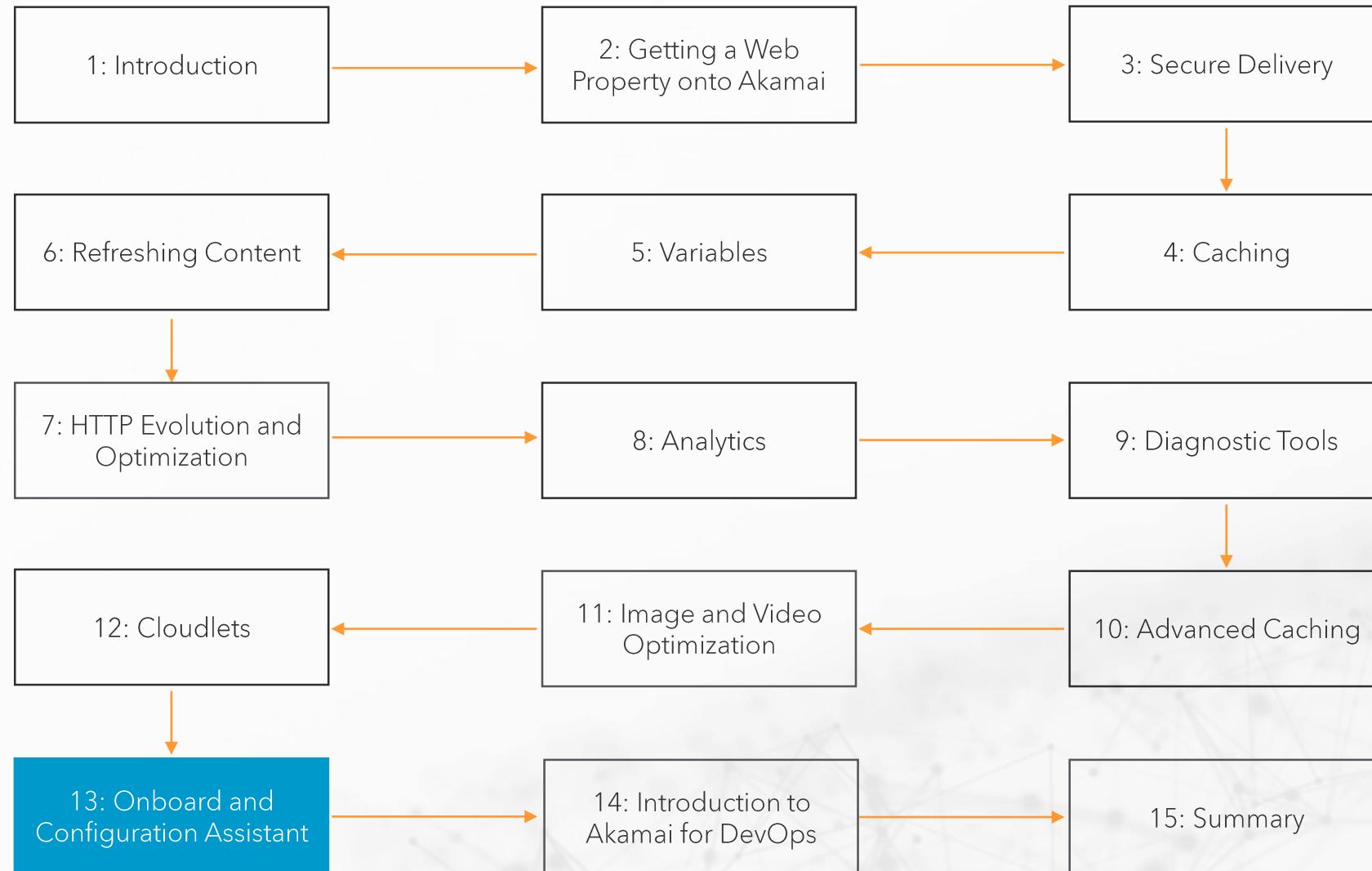
Summary

Cloudlets

- Cloudlets Platform
- Edge Redirector Cloudlet
- Request Control Cloudlet
- Cloudlets Origins
- Application Load Balancer Cloudlet



Agenda



MODULE 13

Onboard and Configuration
Assistant



Module Objectives

At the end of this module, you will understand:

- What OCA is and why you might use it.
- How to setup a basic web property with OCA.
- Limitations of OCA compared to Property Manager.



Onboard & Configuration Assistant

Why OCA?

- Fast, easy and simple to create Akamai web properties
- Simple 4 step process with minimal knowledge required

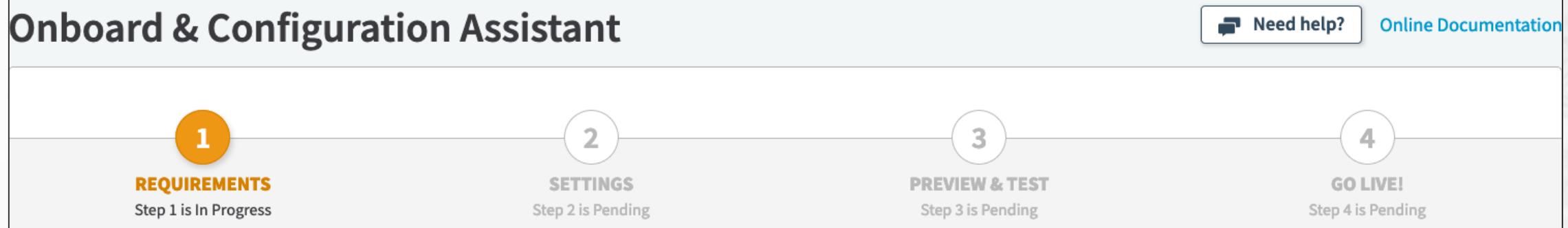
Benefits:

- Guided self-service configuration
- Rapid provisioning
- "Best Practices" recommendation engine

Onboard & Configuration Assistant

Four-Step Self-Provisioning:

1. Site Requirements
2. Configuration Settings
3. Preview and Test
4. Go Live



Step 1 - Site Requirements

Site Requirements

- Specify site hostname(s)
- Choose from default templates for website vs mobile app vs API vs static assets

Your Site Requirements

Welcome!

We will guide you through a series of steps to get you up and running on Akamai. You'll be asked to input your site requirements, then review your configuration settings. You will have an opportunity to preview and test before going live and serving your traffic through our network. If you need to step away, select Exit, and you can pick up where you left off when you return.

Let's get started!

1 Which product do you want to use to onboard your domain?

For now this will be created under contract 1-INC95D.

Product: Ion Standard

Don't see the product you want?
If you don't see the product you're looking for, it may not be supported by your current contract or group.
[Learn more...](#)

2 Enter your primary domain and any aliases.

Primary Domain: www.example.com (Protocols, http:// and https:// are not allowed.)

Aliases (optional): www.example2.com (Protocols, http:// and https:// are not allowed.)

What is the difference between a primary domain and an alias?
An alias domain is a domain that serves the exact same content as your primary domain, using the same origin server. When using an alias domain, your origin sees requests as if they are sent from your primary domain. If you intend to use different origins or serve different content for a domain, it's best to create a separate property rather than adding it as an alias domain.
[Learn more...](#)

3 What do you primarily serve on these domains?

For now we are talking about what you want to serve through our network.

User experience: Website (e.g., desktop only, mobile website only, responsive website)
 Mobile Application (e.g., iOS, Android)
 Application Program Interface (API)
 Static Assets (e.g., images only)

Why do we need to know what you're serving on your domains?
The type of environment that you serve on your domains enables us to determine how requests are sent to your site and how best to configure your settings, such as cache settings, compression, and more.
[Learn more...](#)

4 Do you require your domain(s) to serve secure (HTTPS) traffic through Akamai?

If you are PCI compliant or serve PII (Personally Identifiable Information), you must serve secure traffic (HTTPS) through Akamai's network.

Secure Traffic: Yes
 No

What does it mean to serve secure traffic (HTTPS)?
Serving secure traffic means that you want to support both HTTP and HTTPS traffic. If you are PCI compliant or serve PII (Personally Identifiable Information), you must serve secure traffic (HTTPS) through Akamai's network.
[Learn more...](#)

Step 2 - Configuration Settings

Configuration Settings

- Configure origin hostname
- Either accept the suggested obfuscated hostname, or choose your own

View Options **All Settings** Last modified: Apr 16, 2019 10:33:46 AM **Undo All Changes** **Save**

Your Site Information

- > Site Requirements
- > Origin Information

Your Akamai Configuration Settings

- > Security
- > Performance
- > Intelligence
- > Site Availability
- > Server Offload

Back **Continue to Preview and Test**

Step 3 - Preview and Test

Preview and Test

- Enables your property on the Akamai Network (will take some time)
- Gives you instructions on how to spoof test the site

Onboard & Configuration Assistant

Need help? Exit Online Documentation

1 REQUIREMENTS Step 1 is Complete

2 SETTINGS Step 2 is Complete

3 PREVIEW & TEST Step 3 is In Progress

4 GO LIVE! Step 4 is Pending

We're getting your site ready to preview and test.



We're deploying your site to the Akamai Network. This might take up to an hour. We'll email you when it's complete.

Exit and Email Me When Complete

Step 3 - Preview and Test

Preview and Test

Gives you instructions on how to spoof test the site and test it.

Onboard & Configuration Assistant

[Need help?](#) [Exit](#) [Online Documentation](#)

1 **REQUIREMENTS**
Step 1 is Complete

2 **SETTINGS**
Step 2 is Complete

3 PREVIEW & TEST
Step 3 is In Progress

4 **GO LIVE!**
Step 4 is Pending

Preview and Test

Let's preview and test your site on the Akamai network using one or both levels of testing available. The most basic is a verification test to make sure your site opens. If your site opens, you are good to Go Live.

Optionally, for best results you should go on to fully test your site's functionality. You can conduct this testing now or come back later to complete.

Verify that your site opens

Click the link(s) below to perform a basic verification test to verify that your site opens. If your site opens, you can Go Live now.

Domain [REDACTED]

What do I do if my site doesn't open?

Verify that your origin hostname is set up correctly, your origin server's running, and your origin's SSL certificate passed our checks.

[Learn more...](#)

What if my pages don't render correctly?

For this kind of testing, there's no need for your pages to render. We just want to make sure that Akamai's network was able to reach your origin.

[Learn more...](#)

Step 4 - Go Live

Go Live

Instructs you to CNAME your site to an Akamai Edge Hostname

Onboard & Configuration Assistant

Need help? | Exit | [Online Documentation](#)

REQUIREMENTS
Step 1 is Complete

SETTINGS
Step 2 is Complete

PREVIEW & TEST
Step 3 is Complete

GO LIVE!
Step 4 is In Progress

4

Last step to Go Live!

Before you Go Live, you will need to make changes in your DNS configuration. For each domain, you should replace the existing A record with a CNAME record directing traffic to the specified Akamai domain (e.g. www.example.com.edgesuite.net). Once you've made the change, click Verify DNS Settings to test that the redirects are in place and your settings are correct. When complete, click I'm Finished. After the Time to Live (TTL) for each domain's old A record elapses, you will be live and all of your traffic will be served through our network.

How do I add a CNAME record?
If you have a DNS administrator, send them the information from the table. Otherwise, we can guide you through the process.
[Learn more...](#)

What is my DNS TTL?
The TTL for the CNAME record determines how long it would take to shift traffic back to your origin. After you're comfortable with your configuration, we recommend increasing this TTL to improve performance.
[Learn more...](#)

Verify DNS Settings

OCA Limitations

OCA Overview:

- Production Only (No Staging at this time)
- No APIs for configuration/activation/etc (can still purge with APIs)

Property Manager dependencies:

- You can migrate to Property Manager from OCA
- There is no way back to OCA from Property Manager

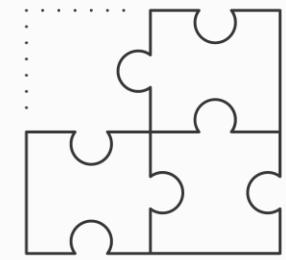
Q&A



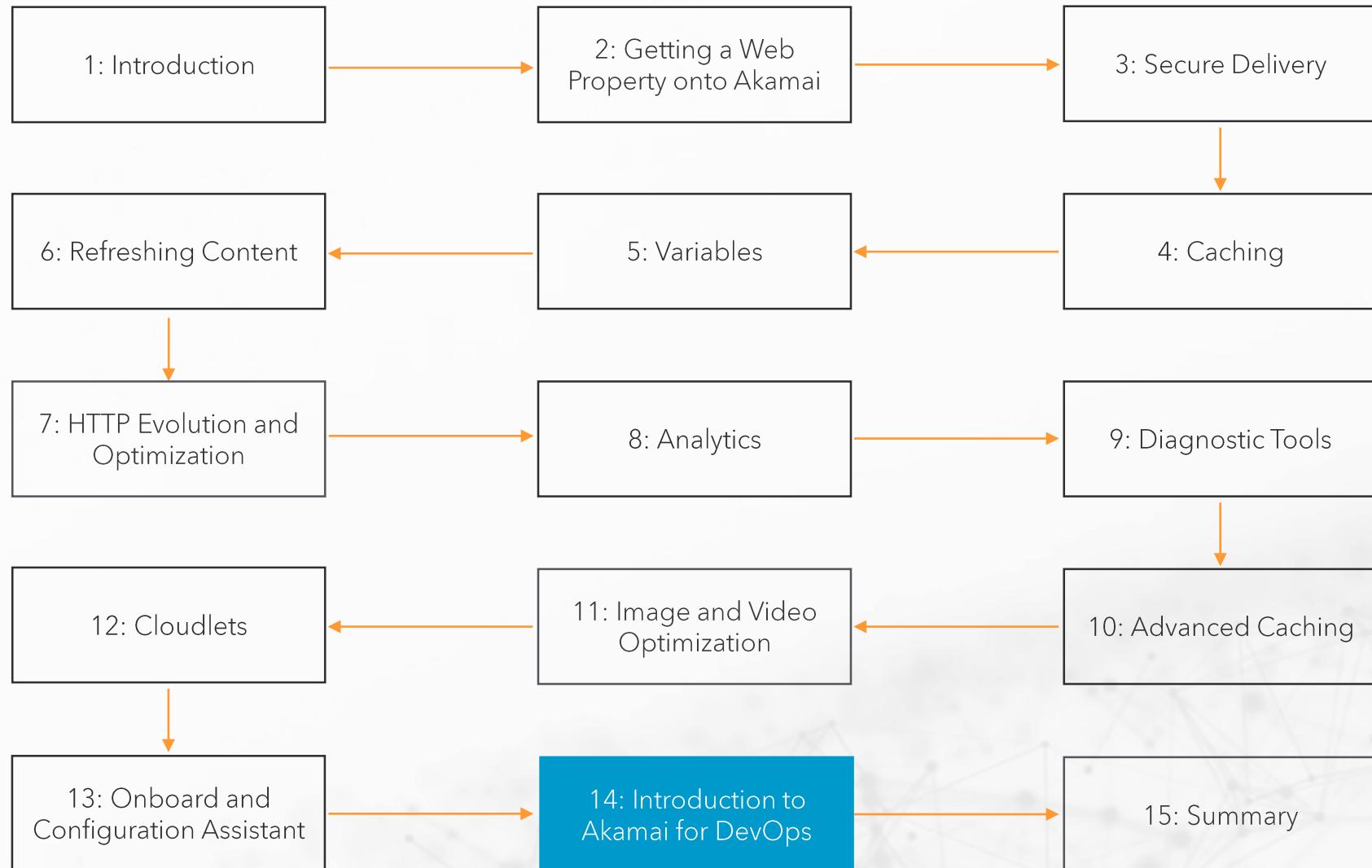
Summary

Onboard & Configuration Assistant

- Intro to Onboard & Configuration Assistant
 - Fast and easy way to get a site onto Akamai
- Using Onboard & Configuration Assistant
 - Four-step process
- Limitations
 - Production network only, no API



Agenda



MODULE 14

Introduction to Akamai for DevOps



Module Objectives

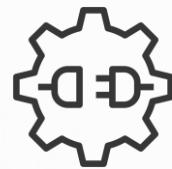
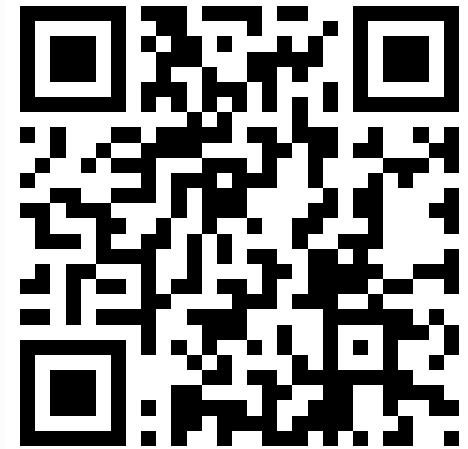
At the end of this module, you will understand:

- Overview of Akamai for DevOps.
- Akamai APIs.
- Akamai CLI (Command Line Interface).
- Mobile App Performance SDK.



Akamai for DevOps

Overview: <https://developer.akamai.com/>



Akamai
API Docs



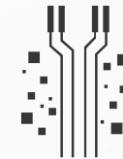
Purge by
Cache Tag



Akamai
mPulse



Akamai
API
Gateway

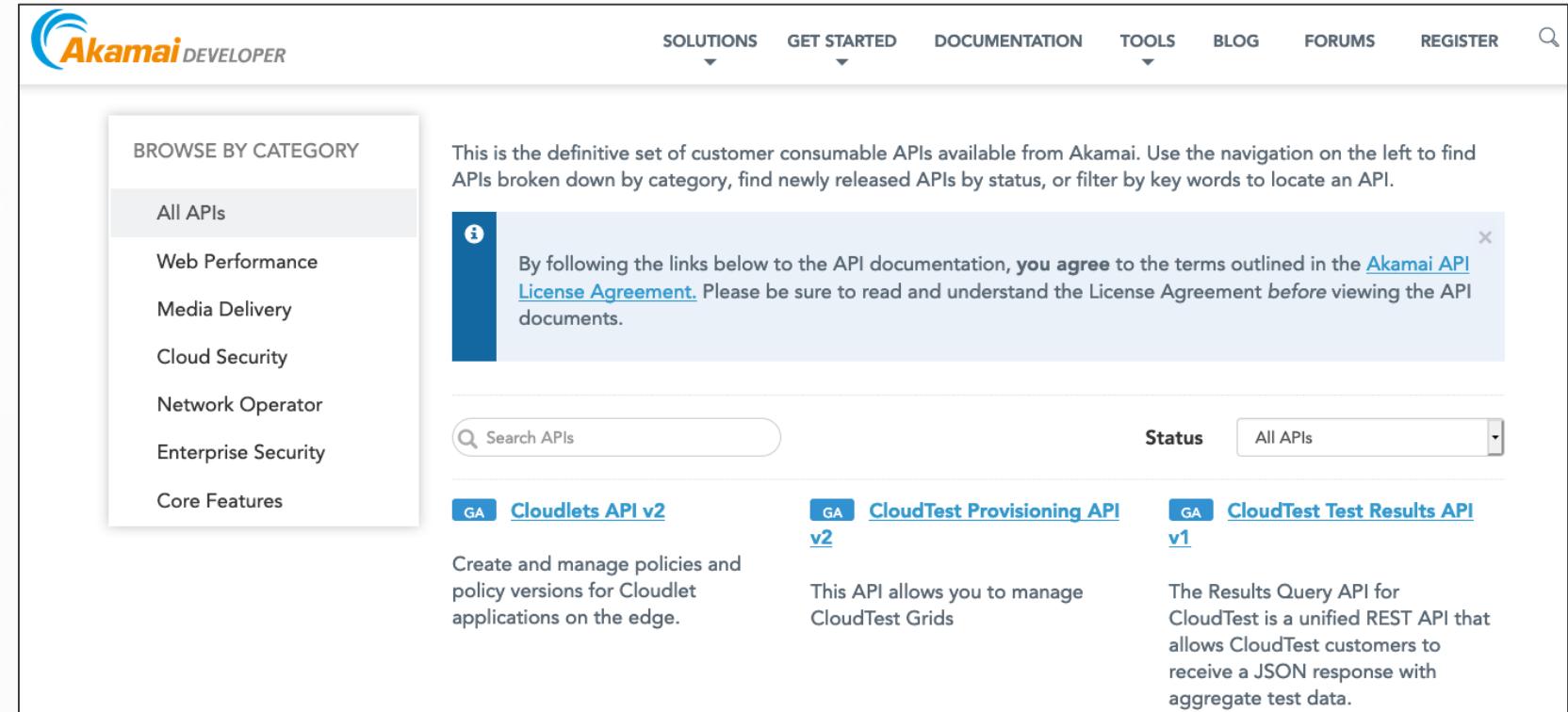


Akamai
DataStream

Akamai APIs

APIs Include:

- Fast Purge (CCUv3)
- PAPI (Property Manager API)
 - API alternative to Property Manager UI
- Image Manager
- Cloudlets



The screenshot shows the Akamai Developer API documentation homepage. The top navigation bar includes links for SOLUTIONS, GET STARTED, DOCUMENTATION, TOOLS, BLOG, FORUMS, and REGISTER, along with a search icon. The main content area features a sidebar titled "BROWSE BY CATEGORY" with links to All APIs, Web Performance, Media Delivery, Cloud Security, Network Operator, Enterprise Security, and Core Features. The main content area contains a message about the definitive set of customer consumable APIs available from Akamai, followed by a note about agreeing to the License Agreement before viewing API documents. Below this is a search bar and a status filter set to "All APIs". Three API entries are listed: "Cloudlets API v2" (GA, v2), "CloudTest Provisioning API" (GA, v2), and "CloudTest Test Results API" (GA, v1). Each entry has a brief description and a link to the API documentation.

APIs Overview: <https://developer.akamai.com/api/>

Akamai APIs

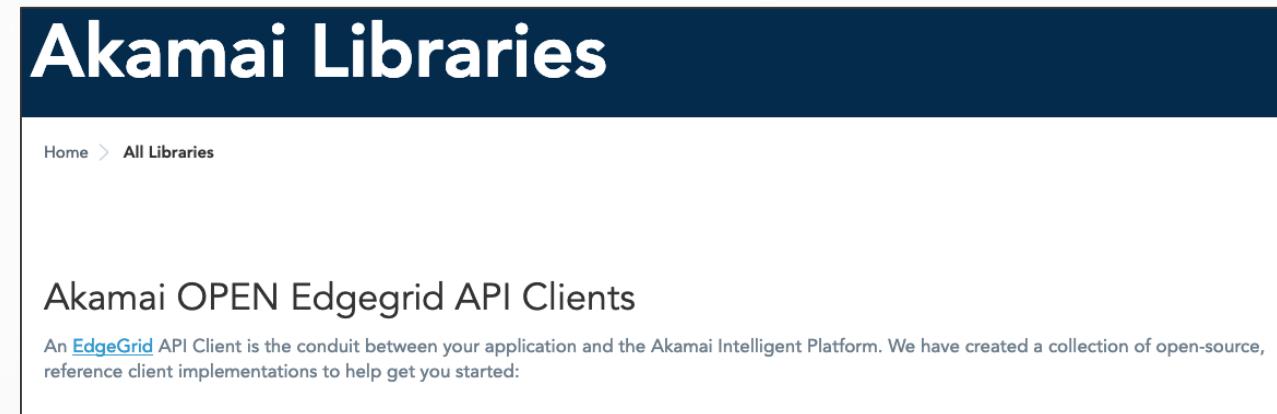
API Authentication - <https://developer.akamai.com/api/getting-started>

1. Create an API Client for a Akamai Portal (control.akamai.com) user.
 - API Client - a derivative user connected to a portal user, which allows API access and define authorizations.
 - API Client contains required tokens to connect to APIs:
 - **access_token, host** - they define access point and access token.
2. Each API Client can contain one or more credentials, which contain additional tokens:
 - **client_secret, client_token** - difference per each credential section.
3. EdgeGrid Authentication Scheme - with all four tokens allow authentication to API (https://developer.akamai.com/legacy/introduction/Client_Auth.html)
 - API client authentication protocol specification for signing HTTP request messages
 - Takes tokens from a file (default: `~/.edgerc`)

Akamai APIs

Client Libraries - <https://developer.akamai.com/libraries>

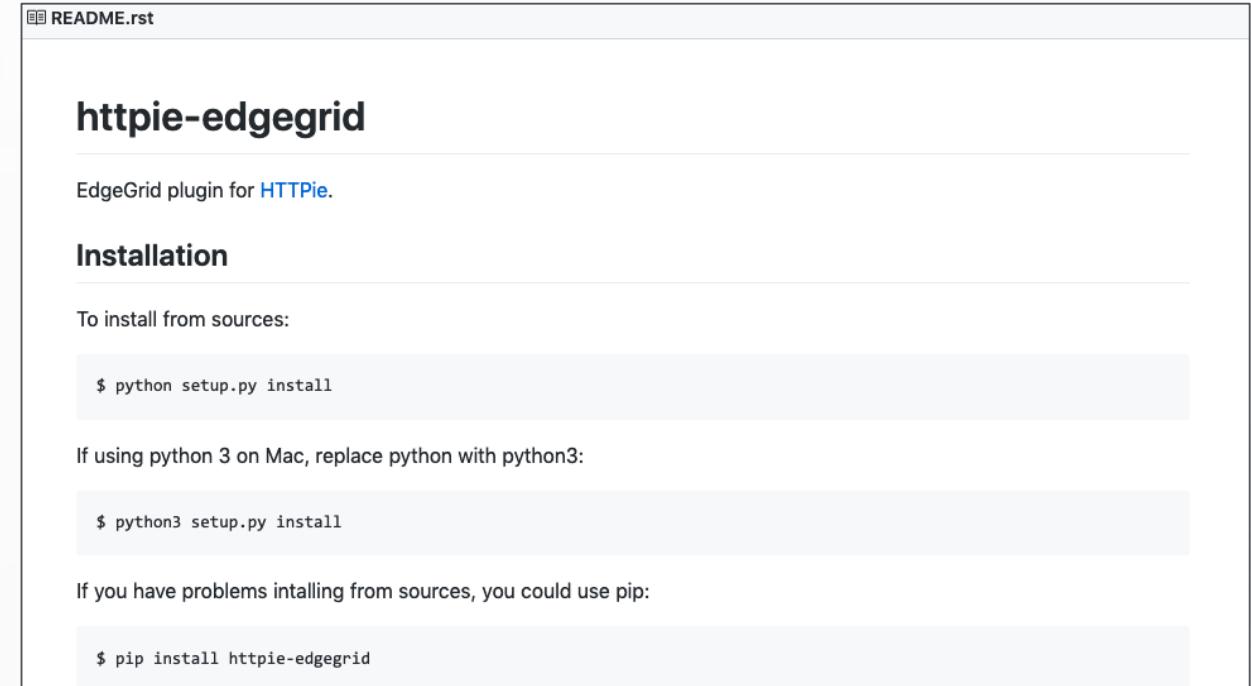
- Implement Akamai EdgeGrid authentication
- Libraries for wrapping API calls
- Languages include:
 - C#, Java, Python, PHP, Golang, Ruby, Node.js, and Perl
- github: <https://github.com/akamai>

A screenshot of the Akamai Libraries website. The header is dark blue with the text "Akamai Libraries" in white. Below the header, a navigation bar shows "Home > All Libraries". The main content area has a white background with the heading "Akamai OPEN Edgegrid API Clients" in large, bold, dark blue text. Below the heading, a smaller text block reads: "An EdgeGrid API Client is the conduit between your application and the Akamai Intelligent Platform. We have created a collection of open-source, reference client implementations to help get you started." The background of the slide features a faint, light gray network or mesh pattern.

Akamai APIs - HTTPie

HTTPie with EdgeGrid:

- <https://github.com/akamai/httpie-edgegrid>
- Great way to get started with APIs
- Simple calls to endpoints



The screenshot shows the `README.rst` page for the `httpie-edgegrid` plugin. The page title is `httpie-edgegrid`, described as an EdgeGrid plugin for [HTTPie](#). It includes an **Installation** section with instructions for installing from sources using `$ python setup.py install` or `$ python3 setup.py install` for Python 3 on Mac. It also suggests using `$ pip install httpie-edgegrid` if there are problems with the source install.

HTTPie - Fast Purge Example

Calling FastPurge API: `http -A edgegrid -a ccu: POST :/ccu/v3/delete/url/staging objects:='["http://demo.aka.akamaiuweb.com/products.php"]'`

```
http -A edgegrid -a ccu: POST :/ccu/v3/delete/url/staging
objects:='["http://demo.aka.akamaiuweb.com/products.php"]'
HTTP/1.1 201 Created
Allow: POST
Connection: keep-alive
Content-Length: 170
Content-Type: application/json
Date: Tue, 09 Apr 2019 12:28:25 GMT
Server: Apache
Vary: Authorization
X-RateLimit-Limit: 35
X-RateLimit-Remaining: 33
X-Ratelimit-Limit-Per-Second: 30

{
  "detail": "Request accepted",
  "estimatedSeconds": 5,
  "httpStatus": 201,
  "purgeId": "f82ec754-5ac2-11e9-8d04-5507824363d4",
  "supportId": "17PY1554812904547598-161432768"
}
```

DEMO

HTTPie



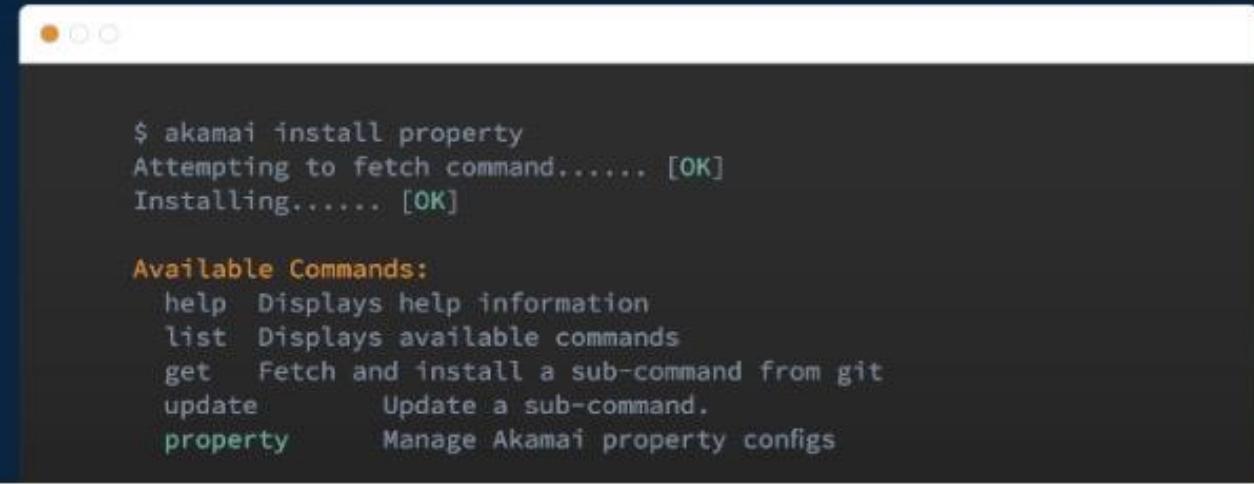
Akamai Command Line Interface (CLI)

Akamai CLI

<https://developer.akamai.com/cli>

Features:

- Modular Command Line Interface
- Open Source (<https://github.com/akamai/cli>)



```
$ akamai install property
Attempting to fetch command..... [OK]
Installing..... [OK]

Available Commands:
  help  Displays help information
  list  Displays available commands
  get   Fetch and install a sub-command from git
  update  Update a sub-command.
  property  Manage Akamai property configs
```

Akamai CLI - Modules

- application security [[appsec](#)] - various commands to control WAF policies
- authentication [[auth](#)] - manipulations on .edgerc file and verification of credentials
- certificate provisioning service [[cps](#)] - manipulation of TLS/SSL certificates deployed on Akamai platform
- fast dns [[dns](#)] - manage Fast DNS zones
- firewall and site-shield [[firewall](#)] - manage firewall notification subscriptions and acknowledge site-shield changes
- image manager [[image-manager](#)] - manages Image Manager policies
- net storage [[netstorage](#)] - Object Store API wrapper for NetStorage product
- property manager [[property](#)] - manage delivery configurations
- property manager [[property-manager](#)] - used when you would like to have a pipeline or federated deployment
- purge [[purge](#)] - manages clearing the cache on Akamai platform
- visitor prioritization [[visitor-prioritization](#)] - this module controls a vp cloudlet policies

Akamai CLI

Akamai CLI – Property

```
sel-mpnab% akamai help property
Usage: akamai property <command> <args> [options]

Commands:
  groups          retrieve account groups
  formats         get rules formats
  search <property>  search for a property name
  create <property>  create a new property
  modify <property>  Modify specified property
  activate <property> activate the property
  deactivate <property> deactivate the property
  delete <property>  delete a property
  update <property>  update target property
  retrieve <property> retrieve rules from target property

Command options:
  --config <config>    Config file          [file] [default: /Users/sakang/.edgerc]
  --section <section>   Config section      [string] [default: papi]
  --debug <debug>       Turn on debugging.  [boolean]
  --help              Show help            [commands: help] [boolean]
  --version           Show version number [commands: version] [boolean]

Copyright (C) Akamai Technologies, Inc
Visit http://github.com/akamai/cli-property for detailed documentation
```

Akamai CLI

Akamai CLI – Purge

```
sel-mpnab% akamai help purge
Usage:
  akamai [global flags] command [command flags] [arguments...]

Built-In Commands:
  config
  help
  install (alias: get)
  list
  search
  uninstall
  update

Installed Commands:
  promotional-deployment (aliases: pd, prodeploy)
  property

Global Flags:
  --bash      Output bash auto-complete
  --zsh       Output zsh auto-complete
  --proxy value Set a proxy to use

Copyright (C) Akamai Technologies, Inc
```

Akamai CLI - Invalidate Example

Using Akamai CLI to Invalidate with Fast Purge:

```
akamai purge invalidate --staging  
http://demo.aka.akamaiuweb.com/products.php
```

Purging..... [OK]

Purged 1 objects (ETA: 5 seconds)

Using Akamai CLI to Delete with Fast Purge:

```
akamai purge delete --production --tag "test" -  
-section ccu
```

Purging..... [OK]

Purged 3 objects (ETA: 5 seconds)

- **akamai** - cli module command line name
- **purge** - module name
- **invalidate/delete** - command name
- **staging/production** - command switches arguments
- **url/tag** - command arguments
- **section** - global tool flag pointing to .edgerc file section

DEMO

Akamai CLI

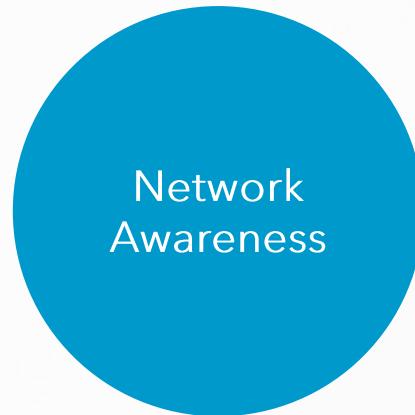


MAP SDK

Mobile App Performance (MAP) SDK: <https://developer.akamai.com/tools/sdk/mobile-application-performance>

Key Features:

- Network Awareness - adjusts the weight of content depending on network conditions.
- Enhanced Performance
 - Universal Caching - reduces requests to the origin, increasing delivery function.
 - SureRoute for Cellular - plans the most efficient path through the mobile network to Akamai's distributed Edge servers.
- Content Pre-Positioning - positions content on users' devices for retrieval in poor network conditions.
- Analytics - analyzes user behaviors under varying network and device conditions.



Augment In-App Processing

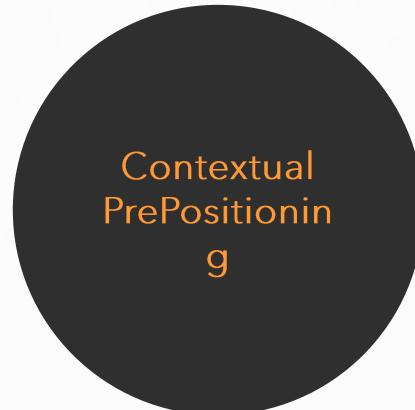
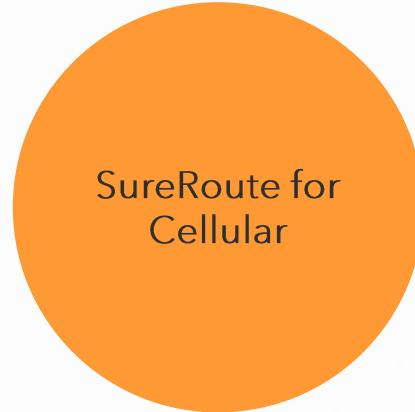
Optimize your app processing based on network quality data, e.g. restrict 3rd party URLs from firing when network quality is poor.

Tight Integration with Existing Akamai Services

Akamai automatically adapts image quality based on client side network quality.

Enable Your API Endpoints to be Adaptive

Utilize your API endpoints to provide responses based on client side network quality, e.g. trim search result payload based on client side network quality.



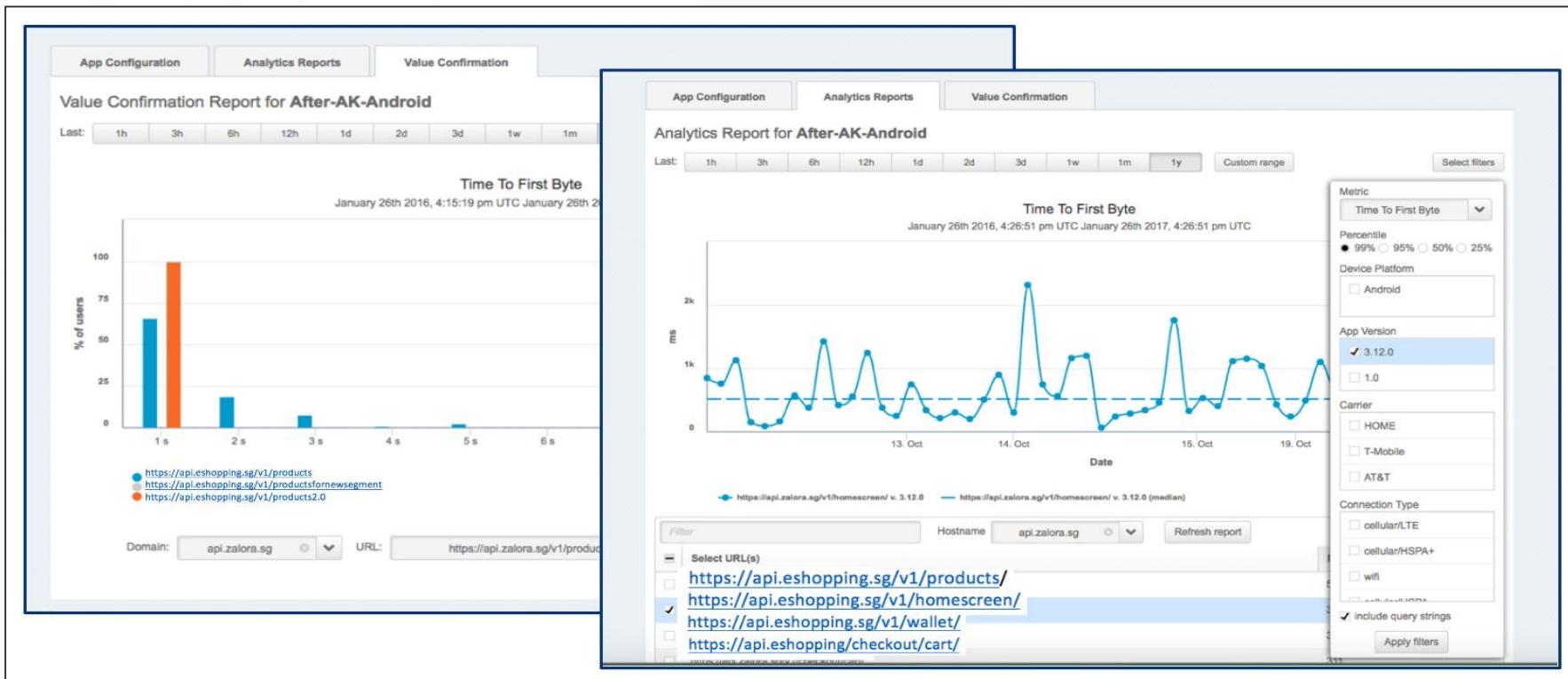
Always Get the Fastest Route over the Cellular Mile

Overcome latency over cellular networks by leveraging Akamai proprietary technology to route your packets through the least congested and fastest region on the Internet.

Instant App Experiences

Preload content into your app in the background to provide instant startup and seamless browsing experience, even when offline.

Mobile App User Analytics



LAB 8

Using the Content Refresh API (CCUv3)



Q&A



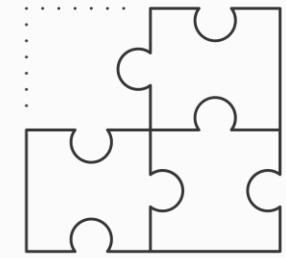
Summary

Akamai APIs

- Using APIs
- Setting up API Client Credentials
- Using HTTPie
- Using Client Libraries

Akamai CLI

- cli-property
- cli-purge



Other

Mobile App Performance SDK

For More Information...

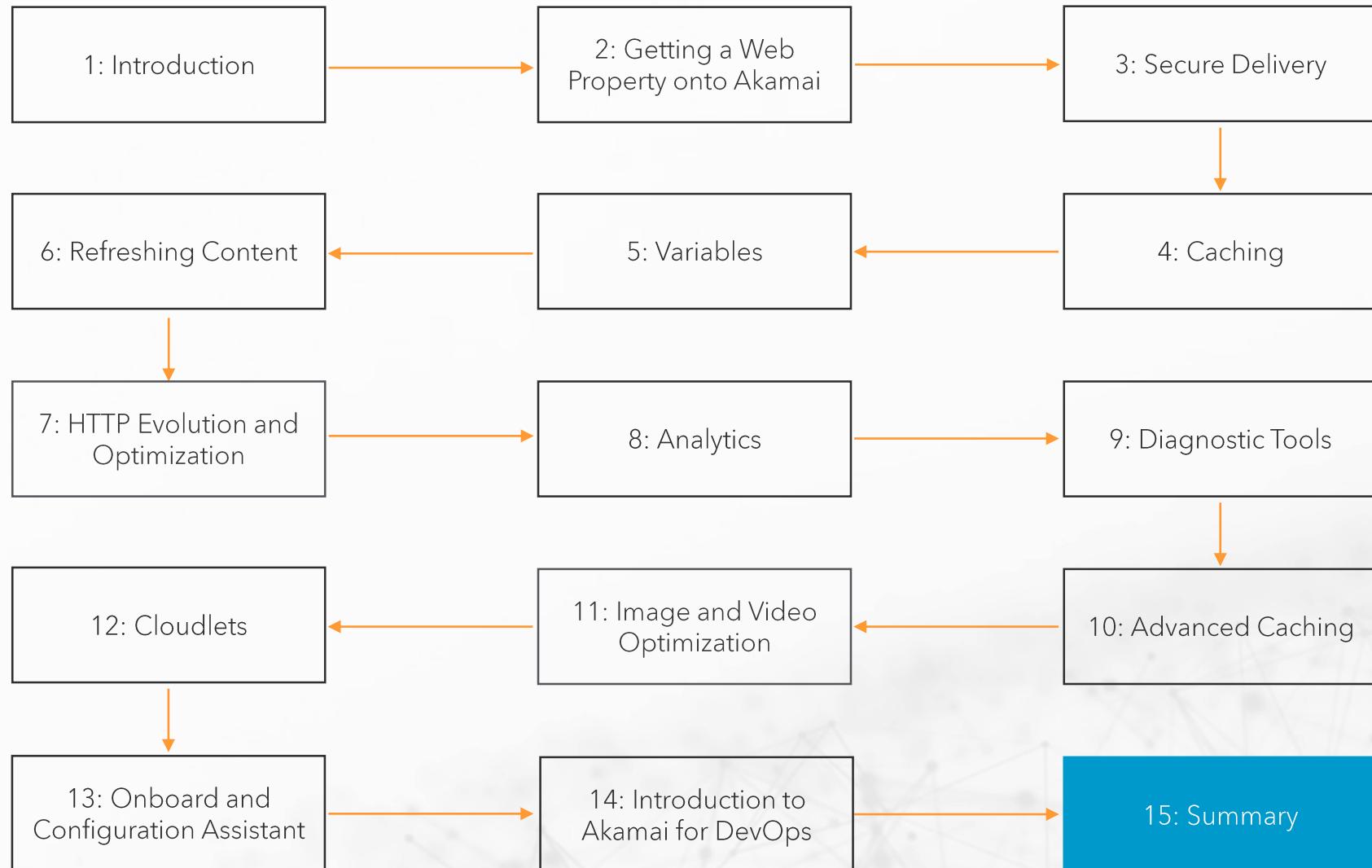
For more info on various Akamai for DevOps topics, see:

- Main Developer Site: <https://developer.akamai.com/>
<https://developer.akamai.com/blog/>
- API Catalog: <https://developer.akamai.com/api/>
 - API Kickstart repo: <https://github.com/akamai/api-kickstart>
 - HTTPie Edgegrid plugin: <https://github.com/akamai/httpie-edgegrid>
- Akamai CLI: <https://github.com/akamai/api-kickstart>
<https://developer.akamai.com/devops/capabilities/#products-akamai-cli>
<https://github.com/akamai/cli>
- Mobile App Performance SDK: <https://developer.akamai.com/tools/map/index.html>

Q&A



Agenda



MODULE 15

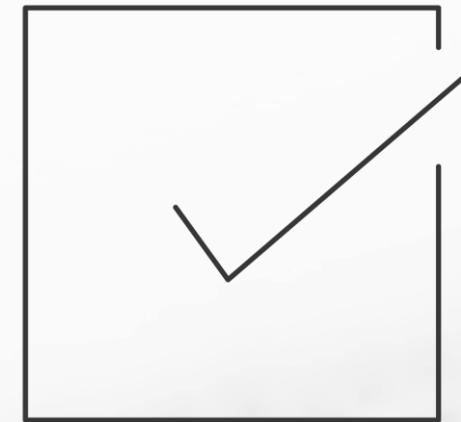
Summary



Congratulations!

Having completed this training, you should now be able to:

- Describe how to get property onto Akamai.
- Explain how secure delivery is configured and realized on Akamai.
- Use Akamai Control Center variables.
- Refresh content on Akamai CDN.
- Tune property caching setup.
- Explain HTTP evolution and optimization.
- Derive insights to your property using analytics features and diagnostic tools of the Akamai Intelligent Edge Platform.
- Configure Cloudlets.
- Apply image and video optimization
- Explain Akamai for DevOps support.



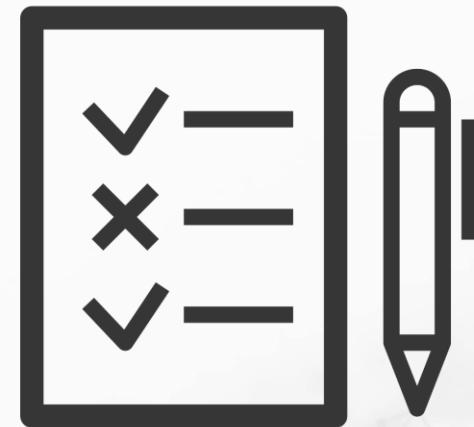
Course Feedback Survey

Tell us what you think about the training! Please take 5 minutes to complete our course feedback survey.



Certification Test

Take 30-45 minutes to complete our certification test. You need to complete the test to receive the certification and the digital badge.



THANK YOU