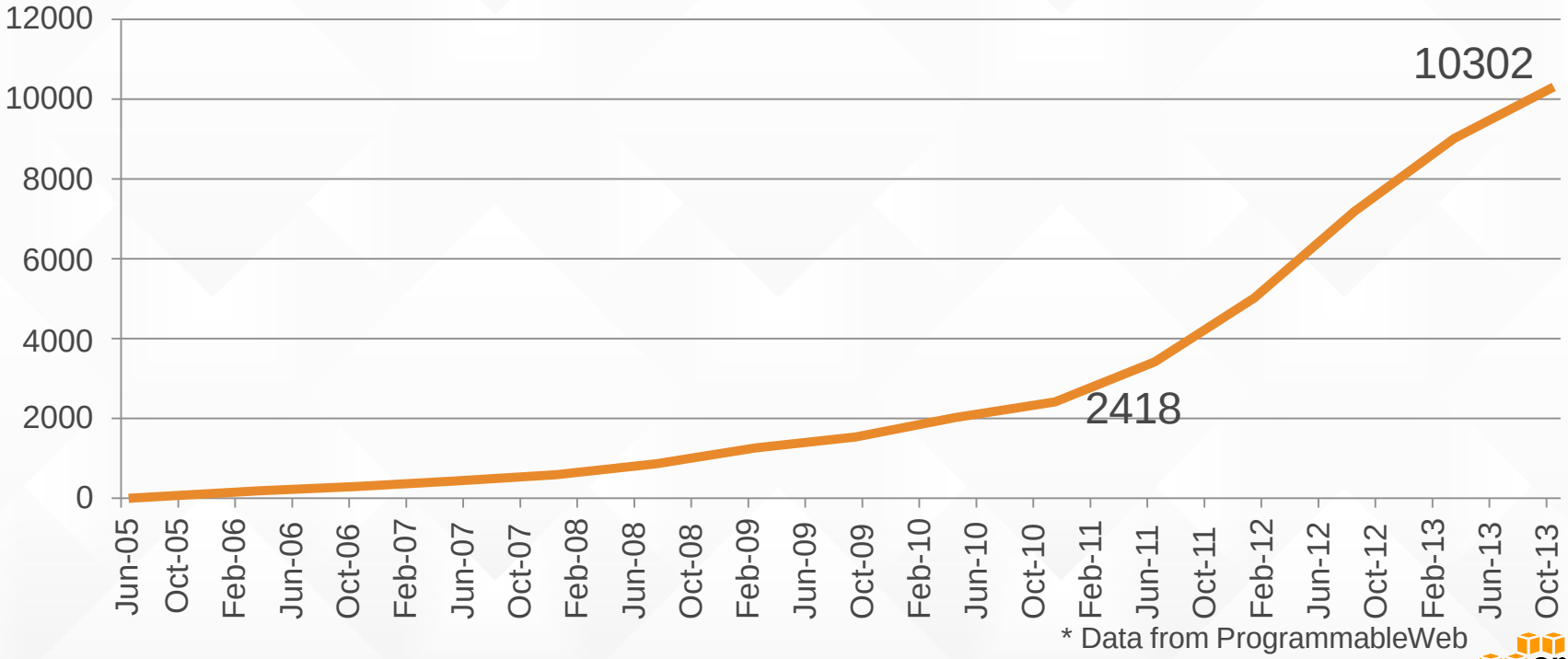
Overview of API Gateway

API proliferation

The number of published APIs is growing rapidly



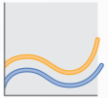
* Data from ProgrammableWeb

amazon web services

- Managing multiple versions and stages of an API is difficult.

- Monitoring third-party developers' access is time consuming.

- Access authorization is a challenge.



- Traffic spikes create an operational burden.

- What if I don't want servers at all?

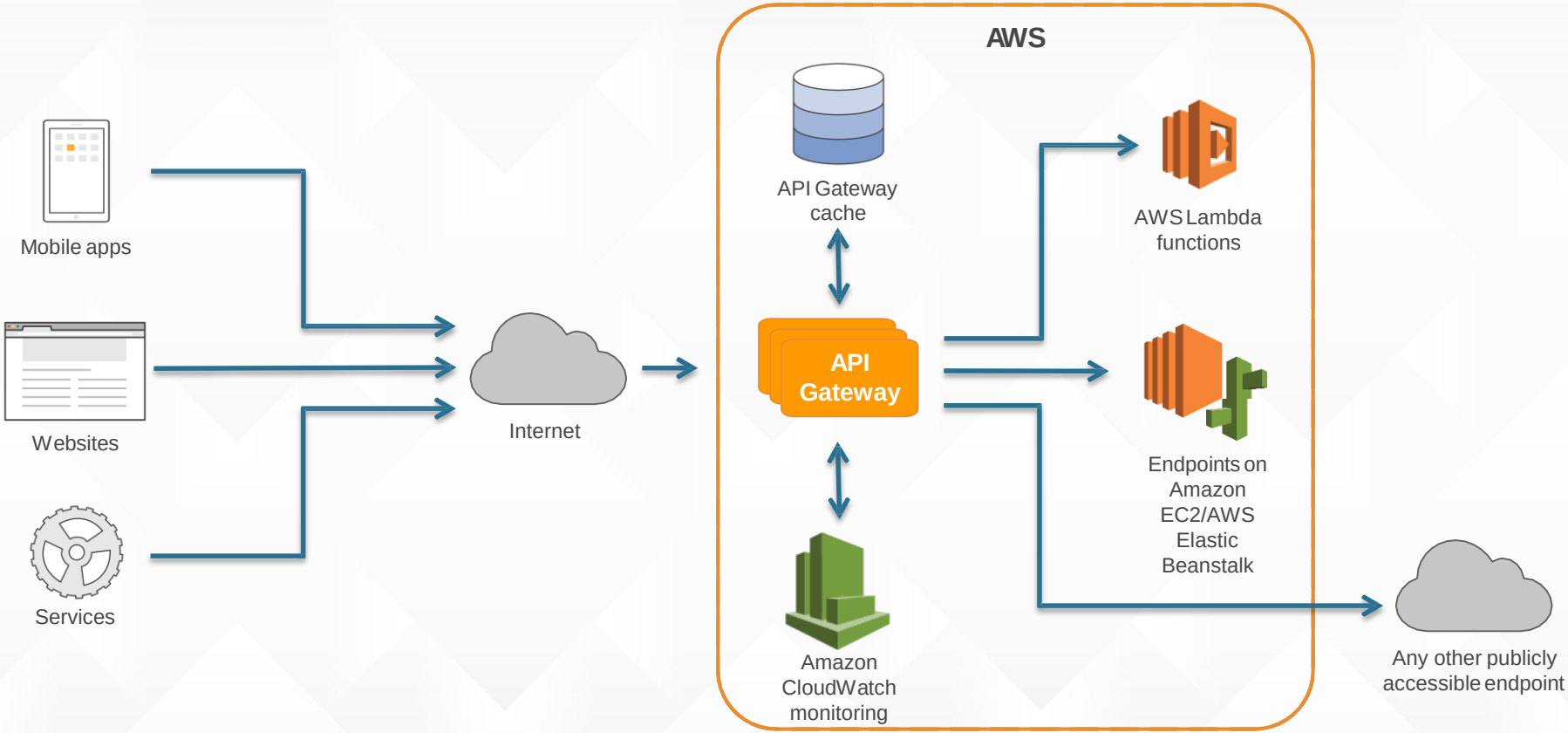
Introducing Amazon API Gateway

- Host multiple versions and stages of your APIs
- Create and distribute API keys to developers
- Leverage signature version 4 to authorize access to APIs
- Throttle and monitor requests to protect your back end
- Utilize AWS Lambda

Benefits of using API Gateway

- Managed cache to store API responses
- Reduced latency and Distributed Denial of Service (DDoS) protection through Amazon CloudFront
- SDK generation for iOS, Android, and JavaScript
- Swagger support
- Request/response data transformation

An API call flow



Build, deploy, clone, and roll back

- Build APIs with their resources, methods, and settings
- Deploy APIs to a stage
  - Users can create as many stages as they want, each with its own throttling, caching, metering, and logging configuration
- Clone an existing API to create a new version
  - Users can continue working on multiple versions of their APIs
- Roll back to previous deployments
  - We keep a history of customers' deployments so they can revert to a previous deployment

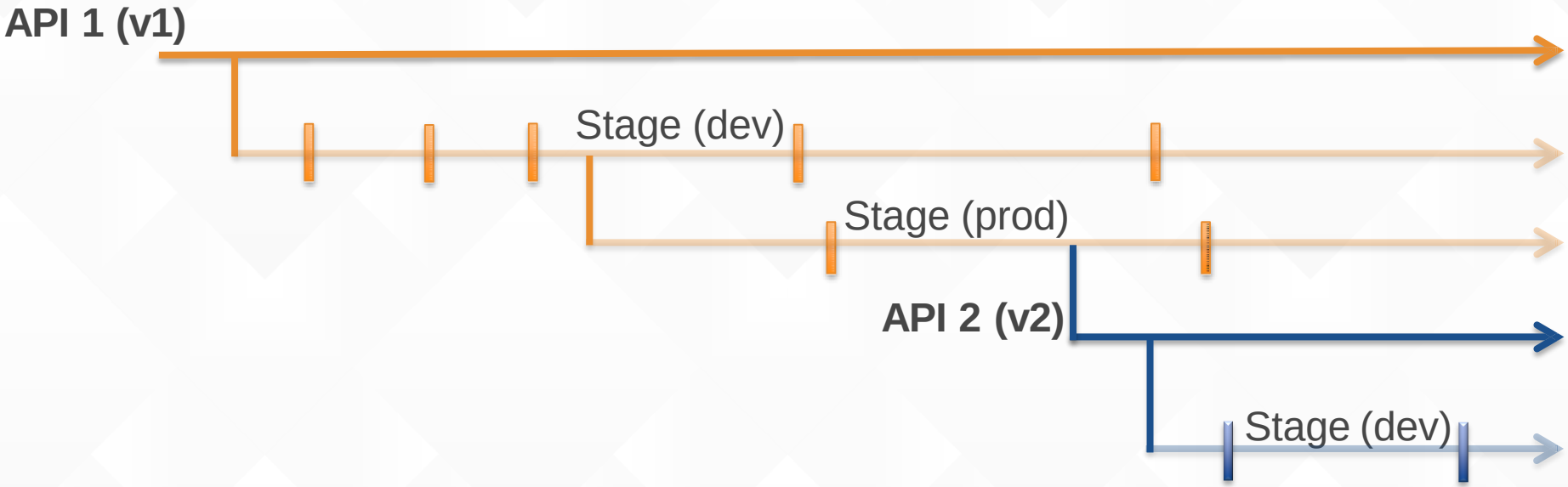
API configuration

- You can create APIs
- Define resources within an API
- Define methods for a resource
  - Methods are resource + HTTP verb

API deployments

- API configuration can be deployed to a stage
- Stages are different environments; for example:
  - Dev (e.g., example.com/dev)
  - Beta (e.g., example.com/beta)
  - Prod (e.g., example.com/prod)
  - As many stages as you need

Manage multiple versions and stages of your APIs



Custom domain names

- You can configure custom domain names
- Provide API Gateway with a signed HTTPS certificate
- Custom domain names can point to an API or a stage
- Point to an API and stage
  - Beta (e.g., yourapi.com/beta)
  - Prod (e.g., yourapi.com/prod)

Use API keys to meter developer usage

- Create API keys
- Set access permissions at the API/stage level
- Meter usage of the API keys through Amazon CloudWatch Logs

Use API keys to authorize access

- The name “key” implies security – there is no security in baking text in an app’s code

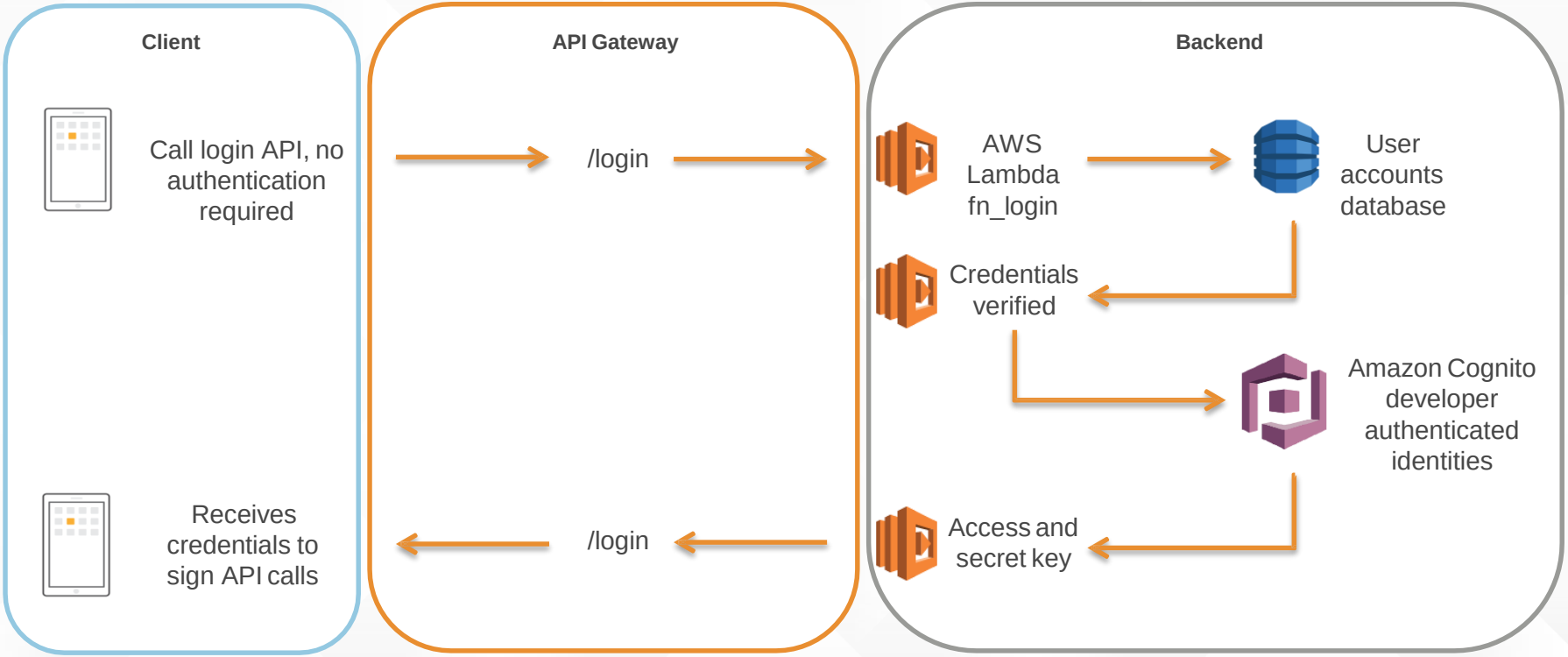
- API keys should be used purely to meter app/developer usage

- API keys should be used alongside a stronger authorization mechanism

Leverage AWS signature version 4 or use a custom header

- You can leverage AWS signature version 4 to sign and authorize API calls
  - Amazon Cognito and AWS Security Token Service (AWS STS) simplify the generation of temporary credentials for your app
- You can support OAuth or other authorization mechanisms through custom headers
  - Simply configure your API methods to forward the custom headers to you back end

Using signature version 4 to authenticate calls to your API




Cognixia
A Collabera LEARNING SOLUTIONS COMPANY
THANK YOU
www.cognixia.com