



# Minneapolis API Management User Group

Aric Day | CA API Management

Aric.Day@ca.com

September 12, 2017

# WELCOME!

# Agenda

- Introductions & Group Goals
- Discussion: Local API Use Cases
- Break ~10:15
- Microservice Security POV
- Product Update – CA Microgateway:Beta
- Closure & Next Session Timing ~11:30

# Companies Represented



**Cargill**™



# Introductions and Group Goals

- Name / Company
- Length of time with products
- Products deployed / Size of deployment
- General functional use cases
- Goals / Outcomes from group

# Around the Room - Topics

- Monitoring | OAuth | API Lifecycle | Architecture
- How do you monitor APIs / What is monitored ?
- Gateway as token server? SAML / OAuth / JWT
- Internal / external developers? APIs exposed internal /external?
- Mobile / Microservice use cases?
- 3<sup>rd</sup> Party Integrations? SSO / Splunk / Ansible / ELK / Tibco

# Follow up Planning

- Locations to host next event ?
- Topics to be covered by clients / partners?
- Preferred time-of-day && day of week
- CA World coming up quickly
  - Las Vegas
  - Expanded education offerings

# The agile business is enabled by architecture

Architectural Maturity

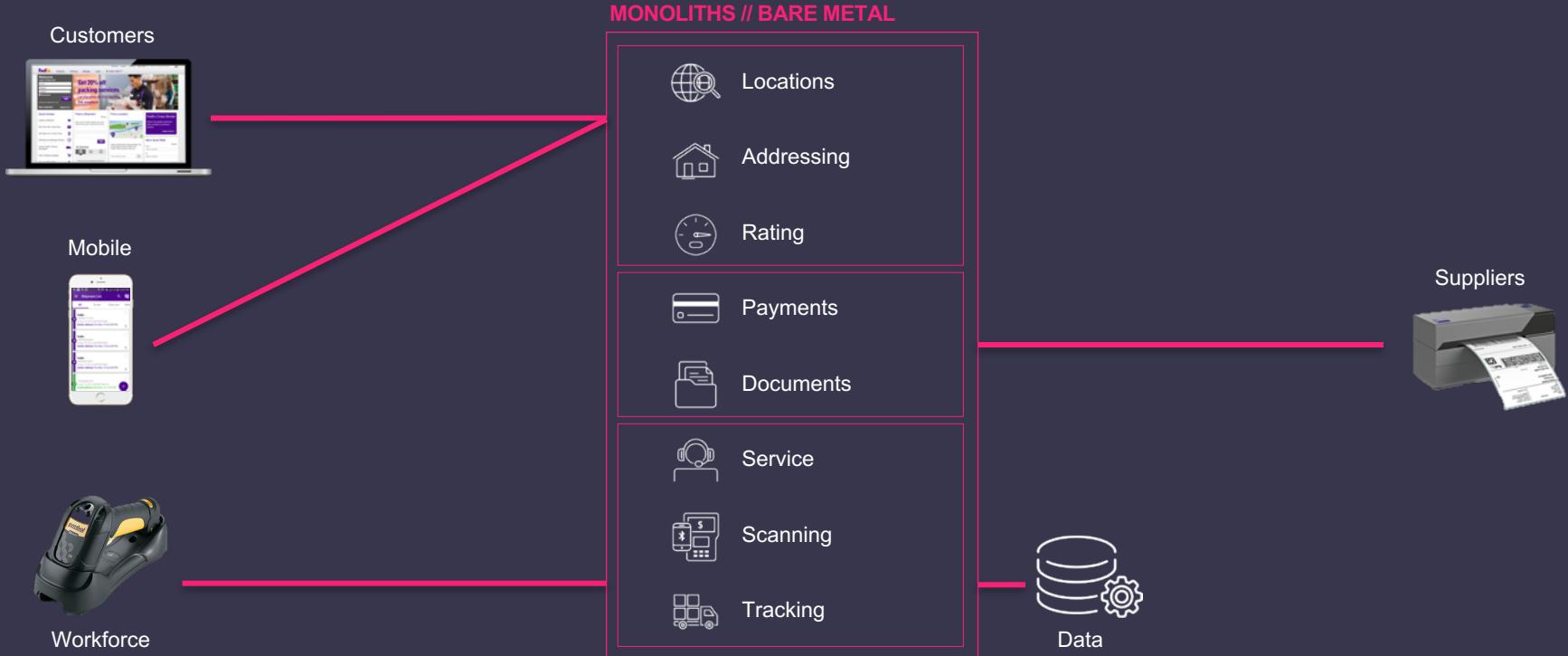
 **Speed** to Revenue

 **Better Experience**

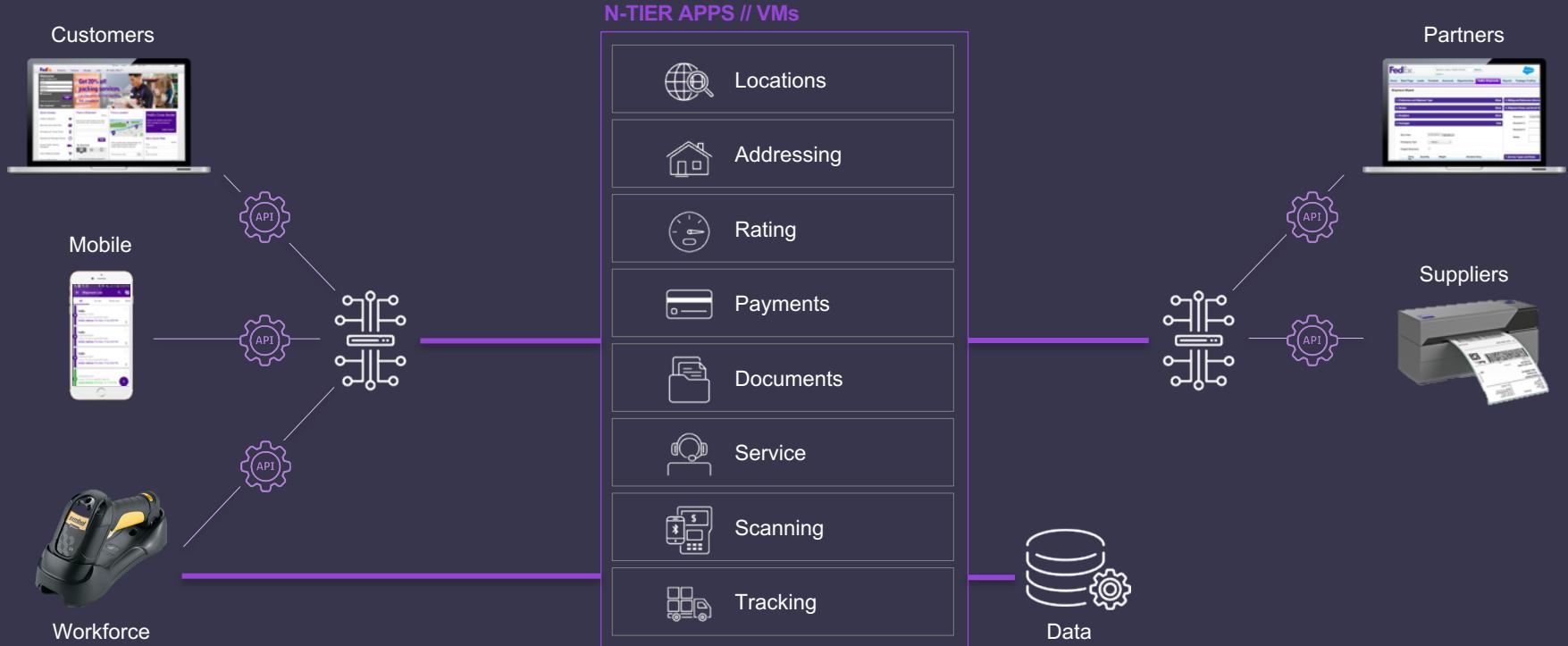
 **Wider Reach**

 **More Innovation**

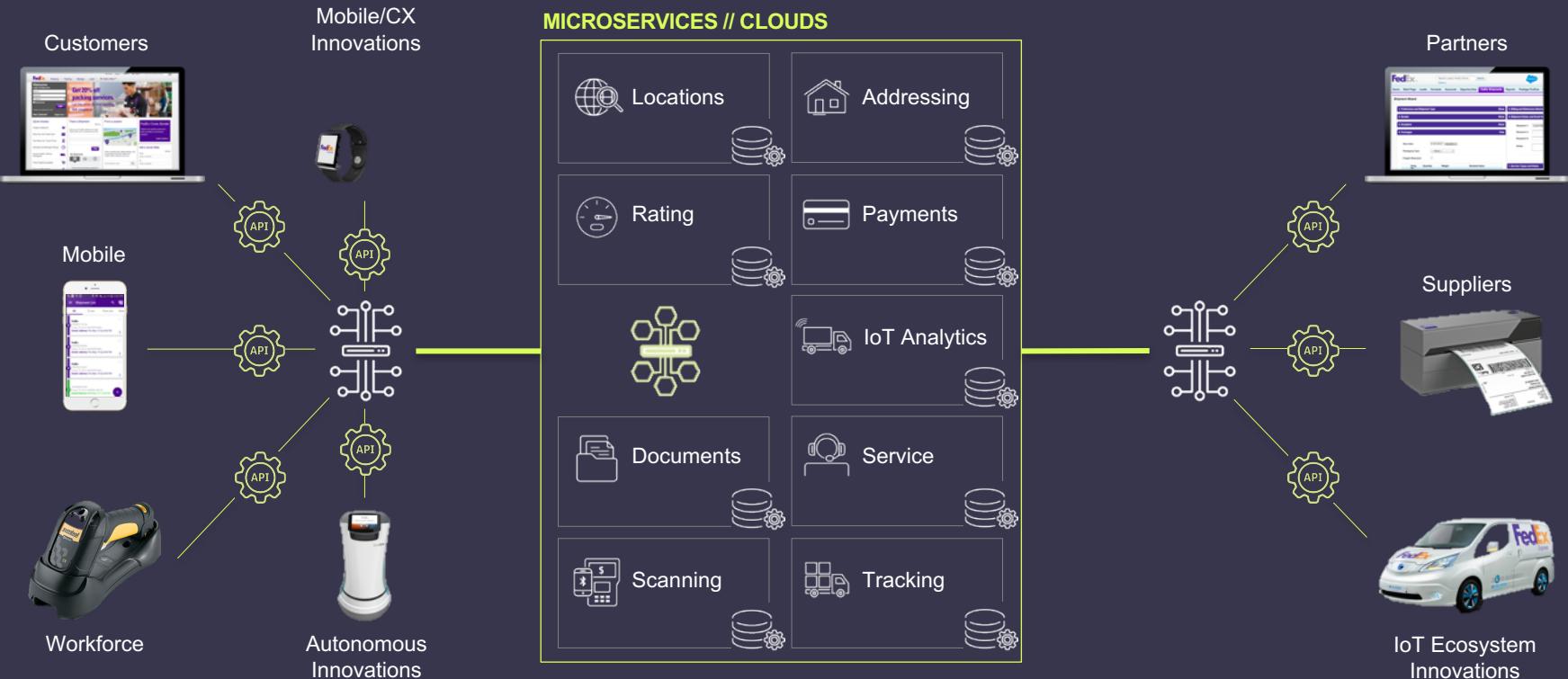
# Monolithic architectures limit agility



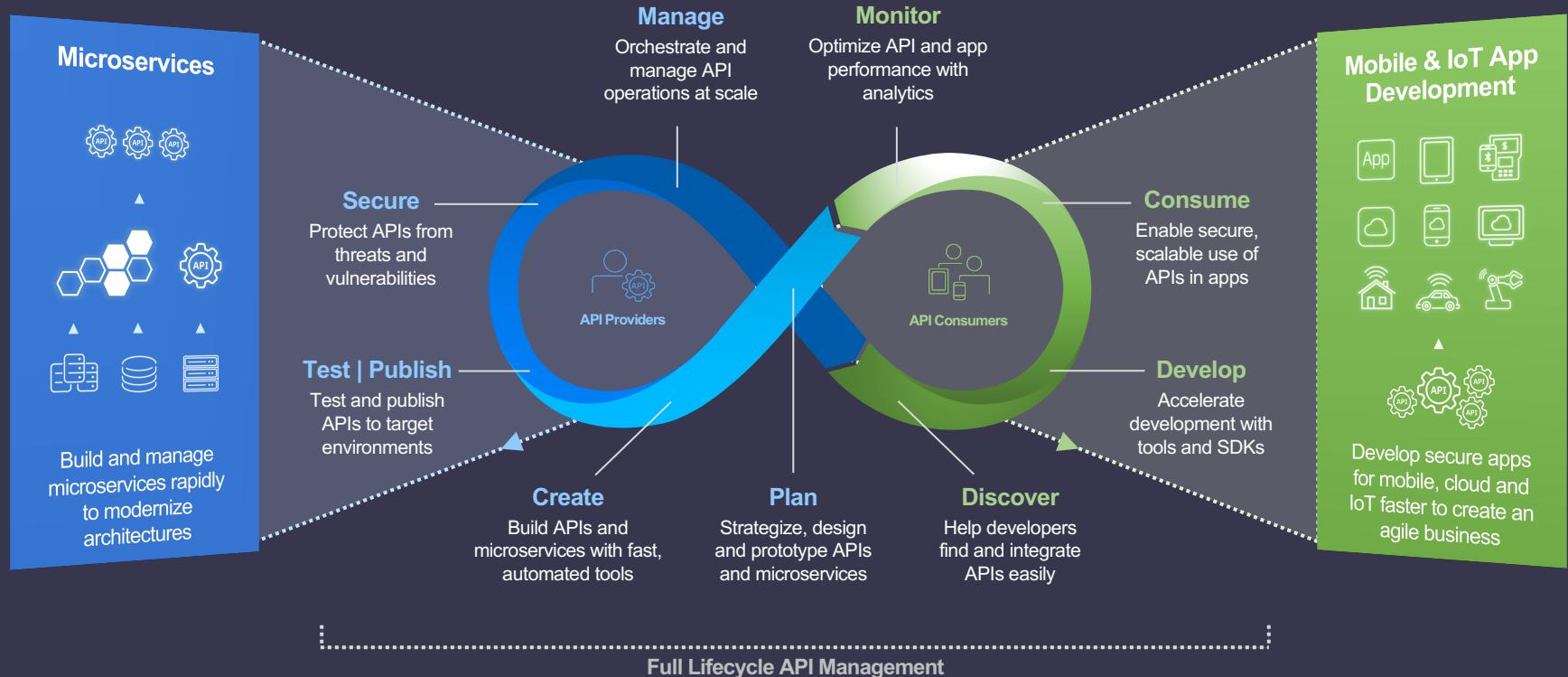
# API-centric architectures improve agility



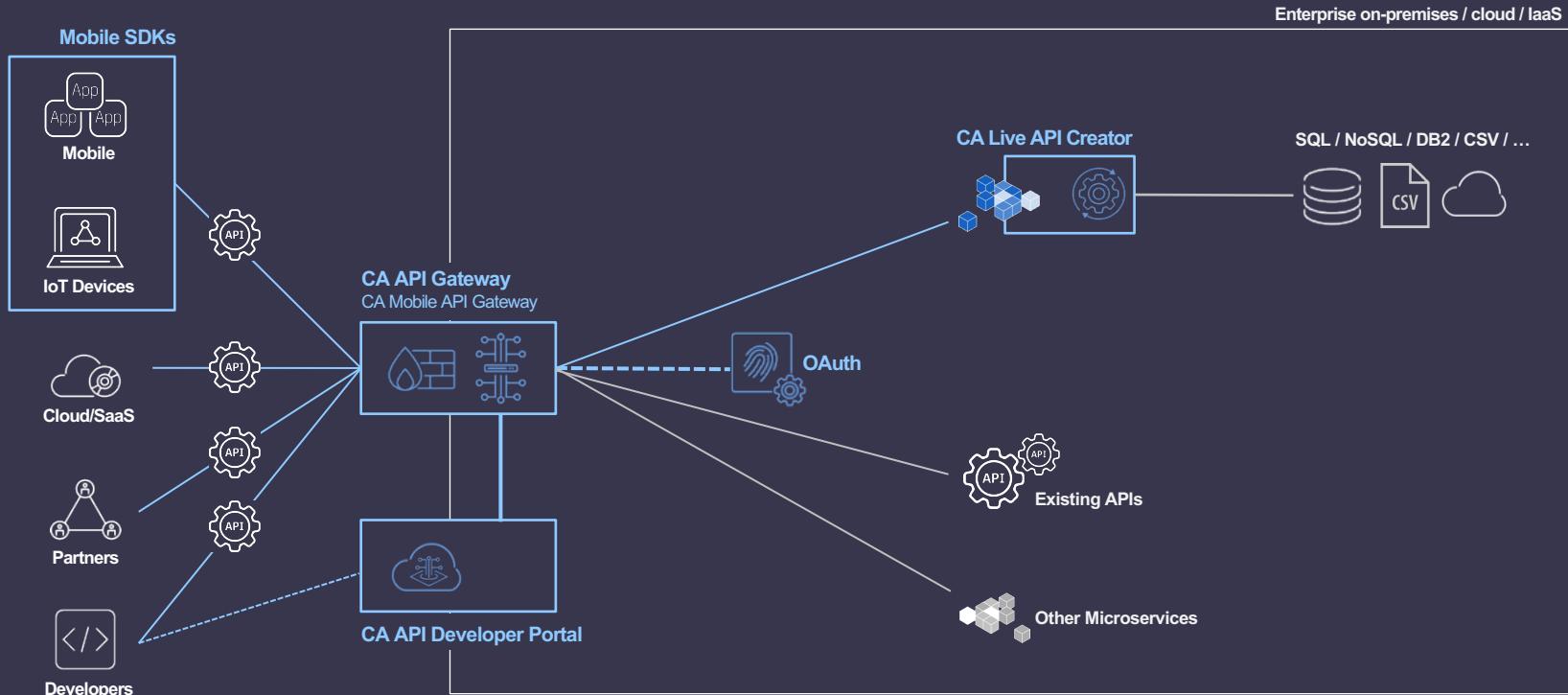
# Microservice architectures **create agility**



# The Modern Application Architecture Model



# Modernizing Architecture with APIs & Microservices



# Strategy and Design

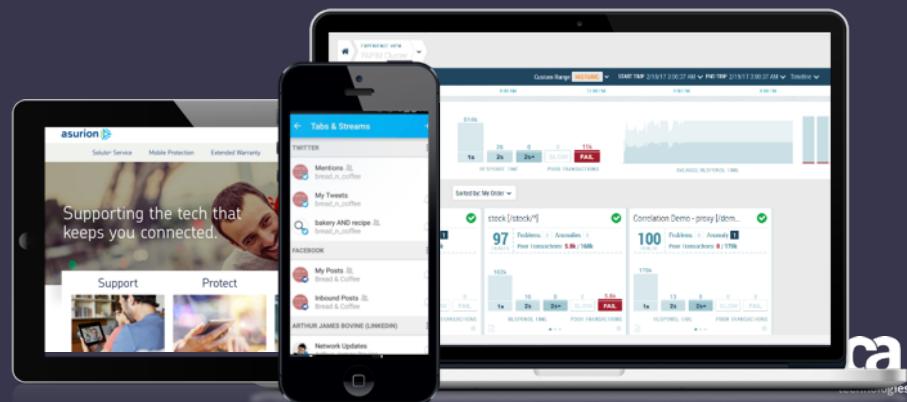
Legacy IT is an obstacle to the rapid and scalable delivery of digital transformation initiatives including mobile, cloud and IoT.

Create an agile business by modernizing or building a new architecture with APIs and microservices. This becomes a foundation to support **digital growth and new business models**.



Our API Academy literally wrote the book on modern microservice architecture.

	
Unparalleled thought leadership and advice from the API Academy	Tools to help prototype and transition to a modern architecture
	
Build architectures that integrate on-premises, public and private clouds	Up to 340% ROI and \$5.9M in net benefits in a typical deployment *



# API & MICROSERVICE CREATION



As the scale of digital transformation increases, it becomes difficult to build and manage APIs and microservices quickly and cost effectively.

Only CA provides a way to create APIs and microservices instantly from existing data. It's **up to 10x faster** than other methods and integrated with the full API lifecycle.



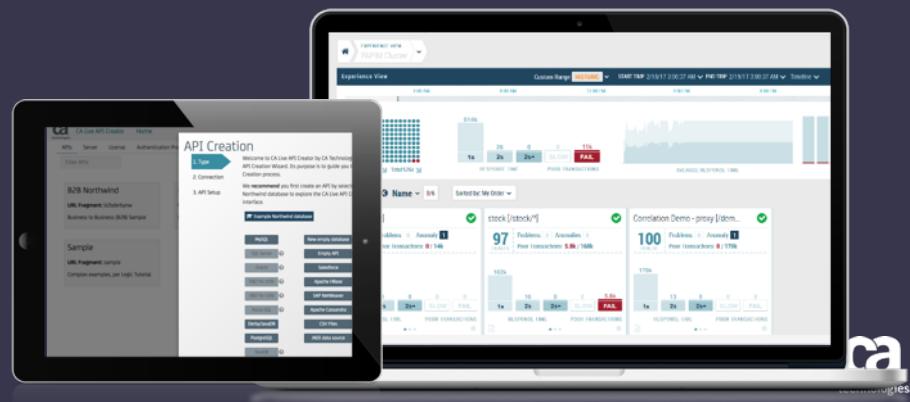
A typical customer spends 50% less time implementing, and 50% less time maintaining their API platform. \*

Connect to legacy data sources from DB2 and SQL to SFDC

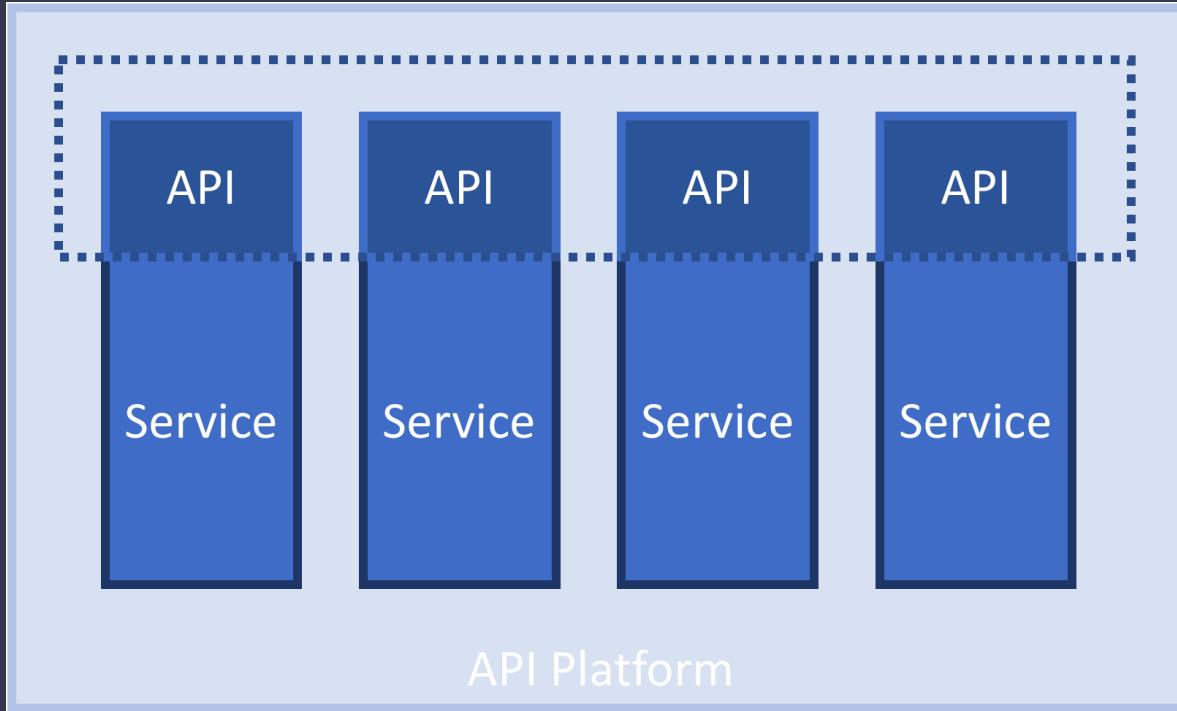
Create microservices and APIs with reactive business logic instantly

The only automated, low-code solution to develop APIs & microservices.

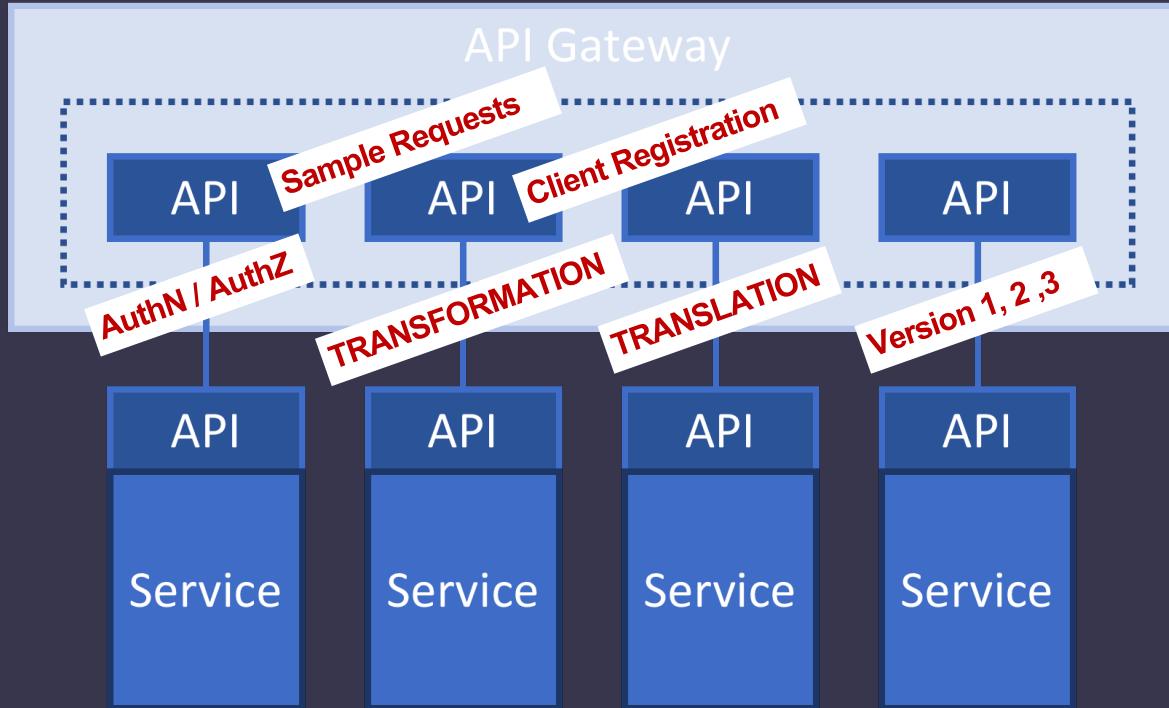
Up to 10x faster and 40x more concise than other approaches



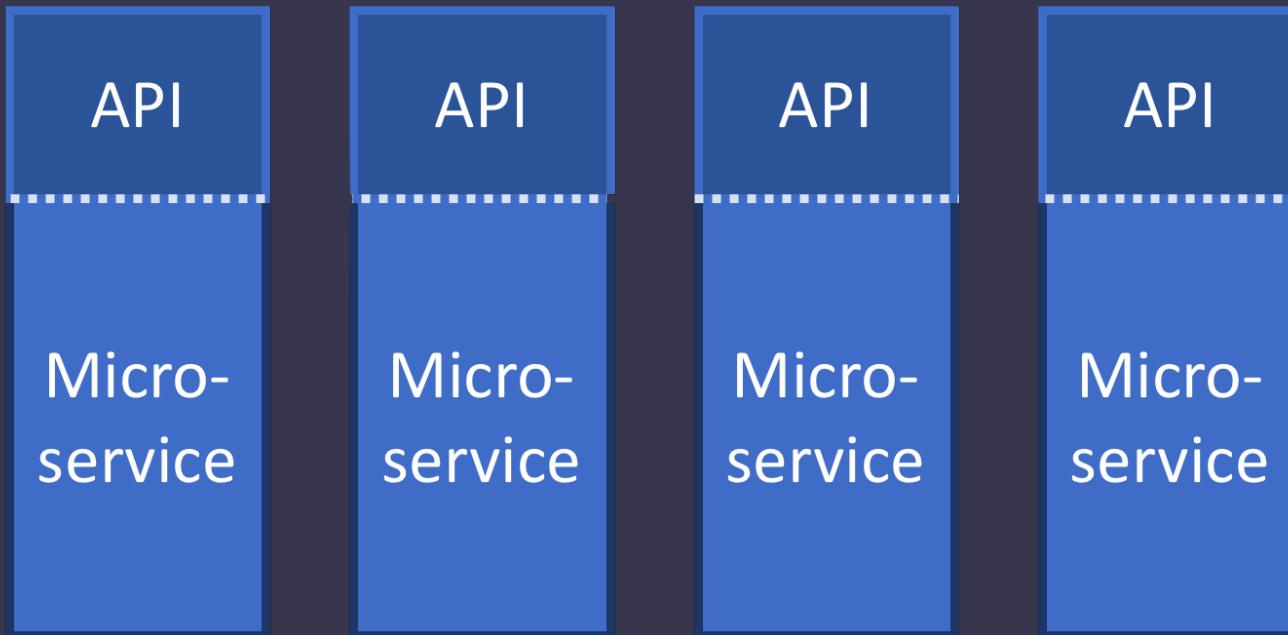
# API Platform - Monolith



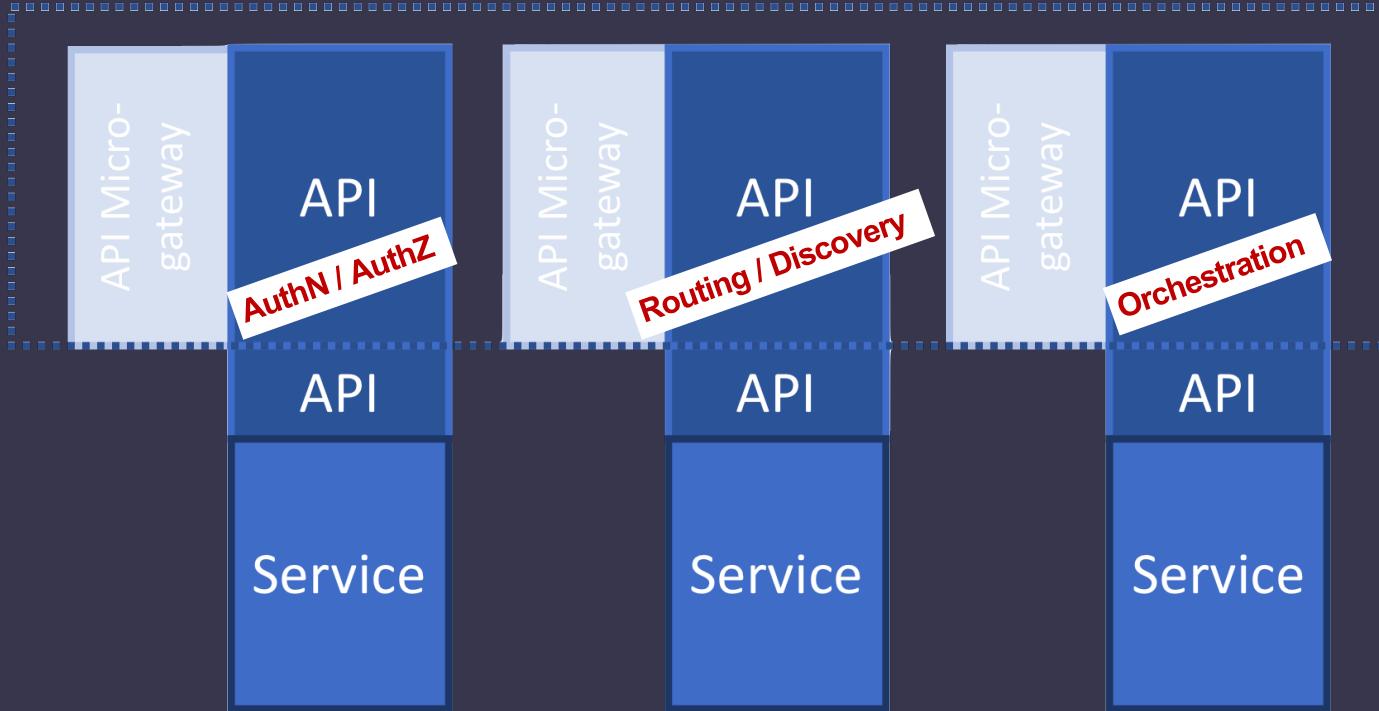
# API Gateway for Monoliths



# APIs and Microservices



# Microgateway for Microservices



# Microgateway – What is it?

CA Microgateway is a new light-weight API Gateway that ...

- Easily and quickly deployed, and managed, as a Docker container using industry-standard tools – Fits CI/CD Pipeline with full automation
- Provides policy templates that enable microservices developers to easily proxy and secure their APIs without needing to learn and use the Policy Manager – JSON configuration
- Provides a policy template framework that allows API Gateway users to create new and extend existing templates
- Complements the microservices capabilities provided by today's other Developer Products, including CA Live API Creator and CA API Gateway

# Demonstration

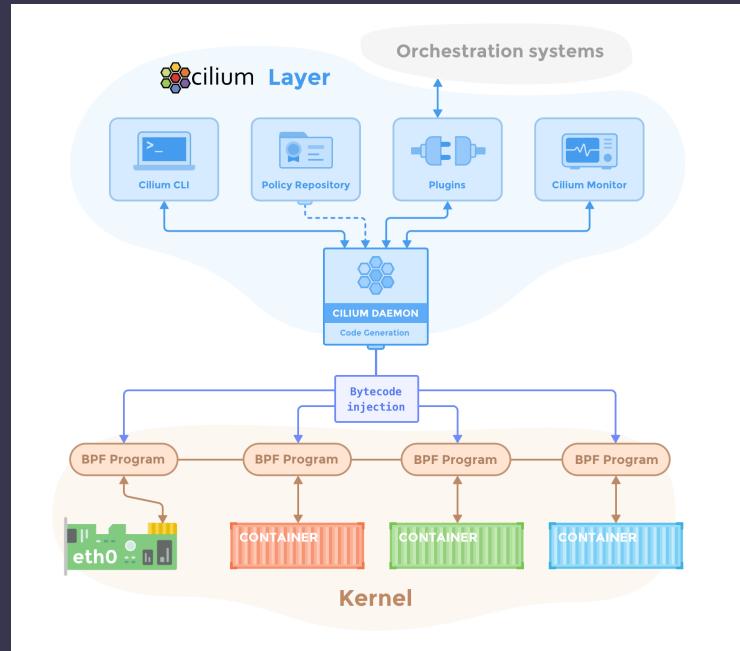
In this demonstration, you'll see how to:

- Configure and start the Microgateway
- Publish an API
- Secure an API
- Build a customized microservices environment
- Showcase end-to-end : Mobile-to-micro

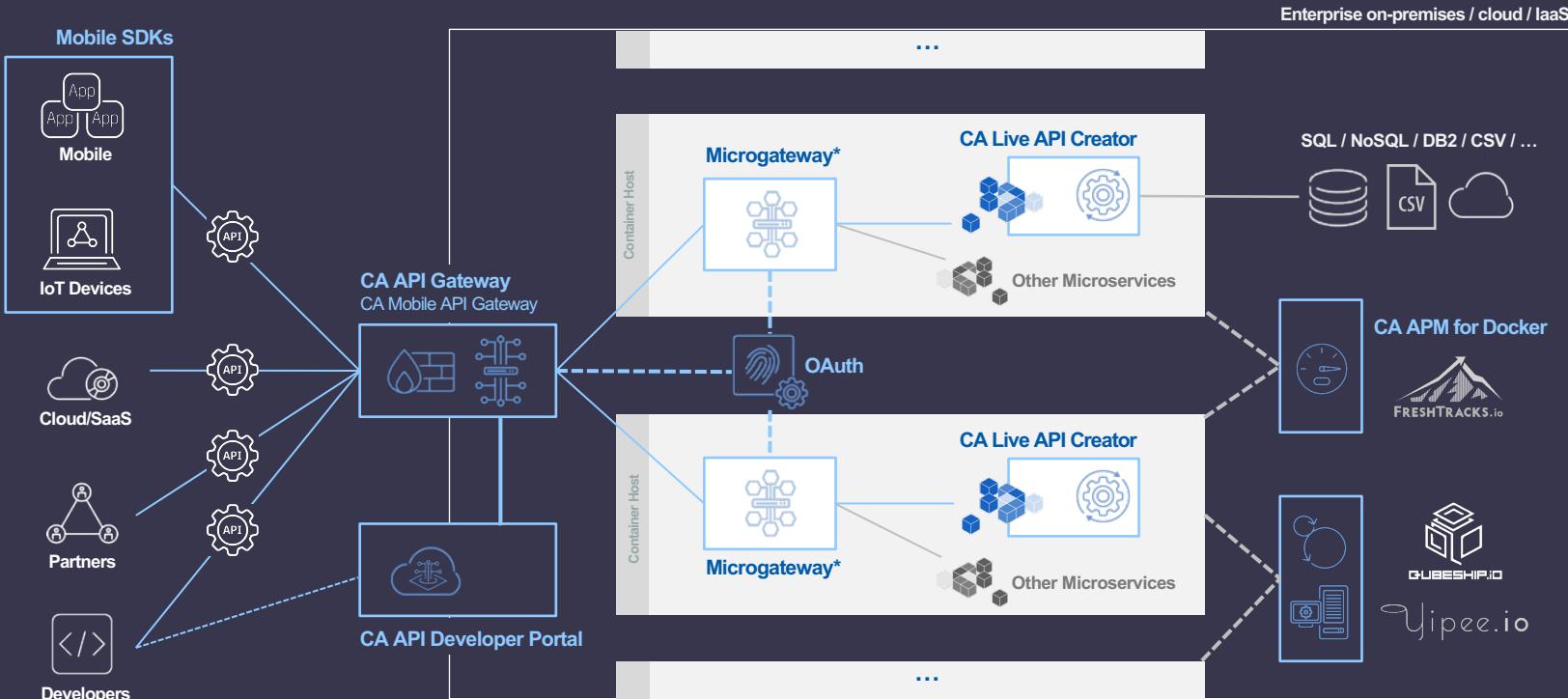


# API Security for Microservices

- Network-based
  - Zone segmentation, ACL's, TLS
- Standards-based
  - OAuth2, OIDC, JWT
- Platform-based
  - AWS, Kubernetes (Istio, Cilium), CF



# Modernizing Architecture with APIs & Microservices



\* A microservices gateway product from CA is currently available in beta. FreshTracks.io, Qubeship.io and Yipee.io are CA Accelerator innovation projects.

# MICROSERVICE ORCHESTRATION

An effective architecture requires visibility, orchestration and consistent security for hundreds or thousands of microservices.



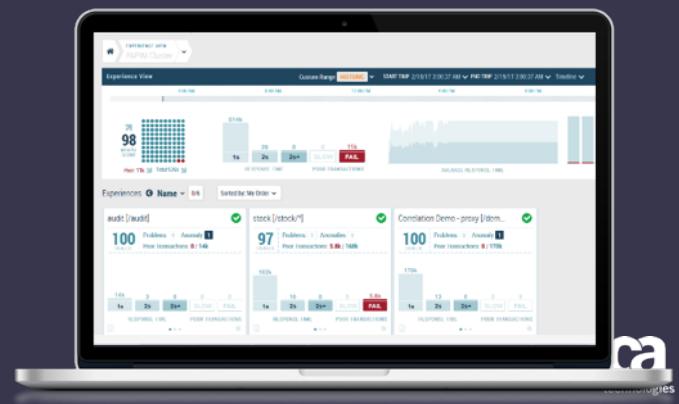
CA offers solutions to **orchestrate and manage the complexity** of microservice architecture both at design time and runtime

Light, DevOps friendly, containerized and ephemeral gateway

Discovery, access control, and routing for microservices

Works with API gateways at the edge for a complete backend

Advanced 360-degree microservices monitoring tools from CA



# Microgateway – Key Characteristics

## Capabilities and Patterns:

- **Service Discovery** and Route via integration with service registries
- **Access Token** for last mile security
- **Rate Limiting** and policy enforcement
- Licensed and priced to **scale** within highly **decentralized** environments
- **Lightweight Orchestration** allows for decoupling of backend services front frontend capabilities
- **Circuit Breaker** to protect from the propagation of failures

## Deployment:

- **Lightweight**, containerized gateway
- Easily **accessible** from common developer platforms
- Deploy and manage using **Docker**
- Incorporate into industry-standard **DevOps** processes
- Support for PaaS environments with **OpenShift** as first target

## Interaction:

- Developer-friendly **policy templates** provided for easy design-time config
- **Extend and enhance** templates to provide custom/**new policies**
- Bake new templates into **reusable containers** for future

# API Gateway vs. Microgateway

	API GATEWAY	MICROGATEWAY
Form Factor	Multiple (OVA, HW, SW, AMI, Azure, Docker)	Docker
Lifecycle	Long running instance, use GMU for migration	Ephemeral, short-lived
Capabilities	Full-Featured	Limited Capabilities focused on microservices patterns
Licensing	Per-Instance model	Sold in packs of instances
Network Topology	Network Edge, one environment for all services	Inside microservices environment, one or few services per microgateway



# MOBILE & IOT DEVELOPMENT

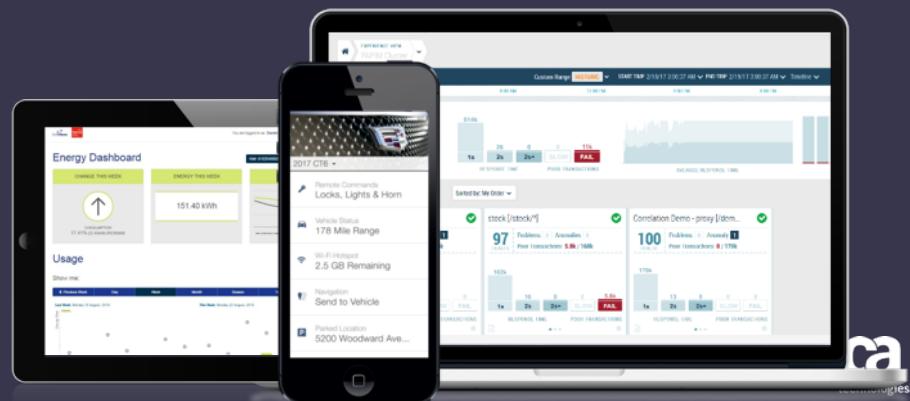


Mobile and IoT applications that consume APIs can be difficult to develop, secure, and manage consistently at scale.

**CA extends across the lifecycle to include robust development tools** like Mobile SDKs and Mobile App Services, for a truly integrated digital transformation solution

A typical customer experiences a 97% reduction in time to market for apps with complex integrations\*

A range of developer tools, SDKs and mobile app services	Mobile SDK and Oauth toolkit to ensure security compliance
Mobile App Services reduce development time of core capabilities	Support for iOS, Android, Cordova, MQTT for a wide range of use cases



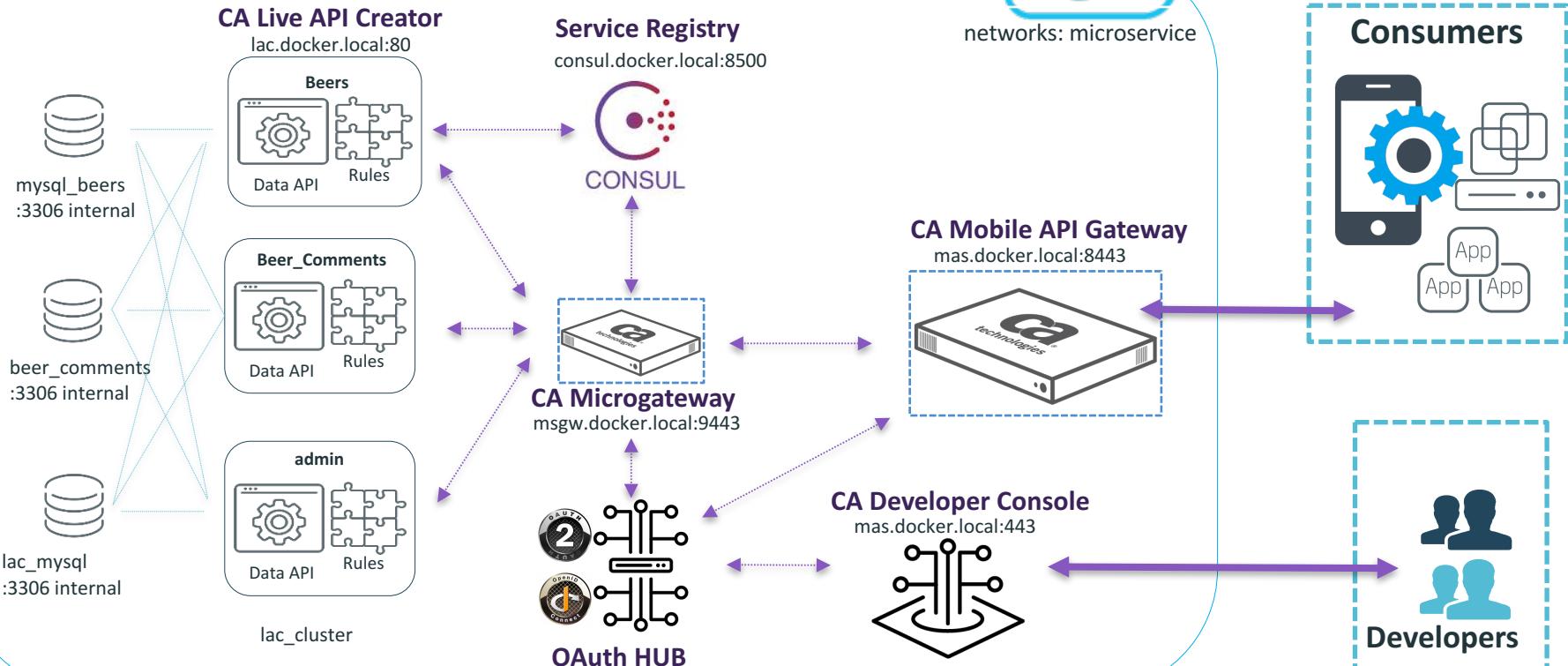
# Demonstration

In this demonstration, you'll see how to:

- Configure and start the Microgateway
- Publish an API
- Secure an API
- Build a customized microservice environment
- Showcase end-to-end : Mobile-to-micro



# Demo Architecture



# Latest Innovations - CA API Management



**Microservices Design:** The API Academy released an open-source tool for rapidly brainstorming and sketching concepts for APIs and microservices



**Microservices Development:** CA Live API Creator delivers the industry's first and only automated, point-and-click solution for rapidly generating complete microservices



**Microservices Deployment:** Support for Docker® deployments ideal for those seeking the most complete API and microservices management solution



**API Portal Architecture:** Newly architected CA API Developer Portal eases deployment and migration to any cloud. New API-first architecture simplifies management, policy configuration, and integration



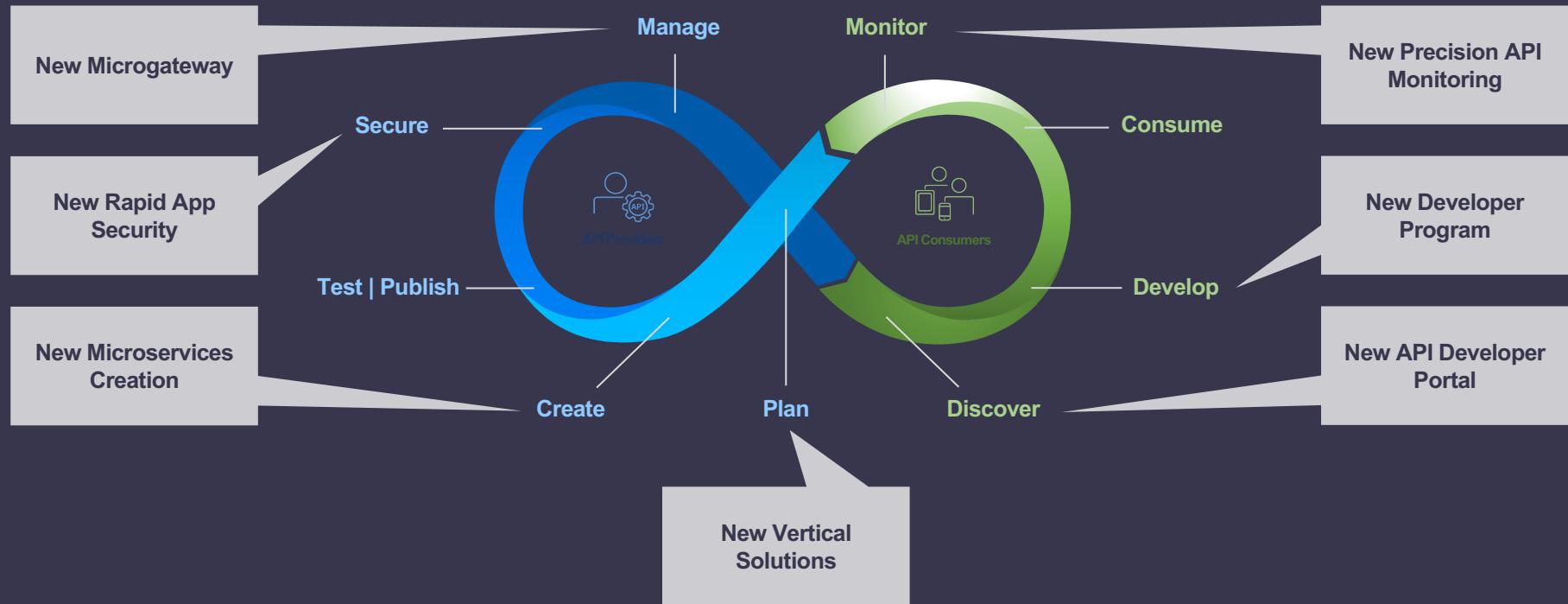
**API Precision Monitoring:** New CA API Management and CA APM solution provides real-time and deep visibility into critical API performance data with transaction tracing, to enable rapid triage



**Mobile App Development:** New contextual security capabilities with CA Rapid App Security accelerates cross-platform mobile development. New Integration



# Latest Innovations - CA API Management



# Thank You.



## Aric Day

Principal Consultant | API Management  
Aric.Day@ca.com

 @cainc

 slideshare.net/CAinc

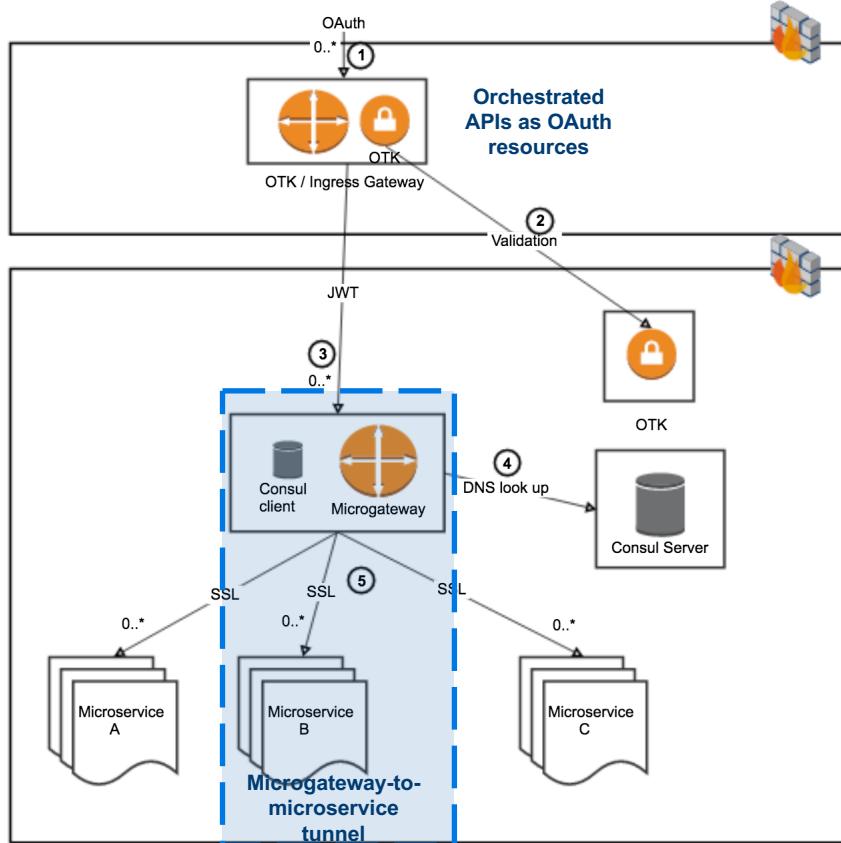
 linkedin.com/company/ca-technologies

**ca.com/API**

# Appendix



# Reference deployment - conceptual



# Reference security model - conceptual

Ingress  
(OTK and MAG)

- Orchestrated API exposed as OAuth resources
- Delegate OAuth token validation to OAuth server
- Convert token to JWT for validated requests to access microservices APIs exposed on Microgateway

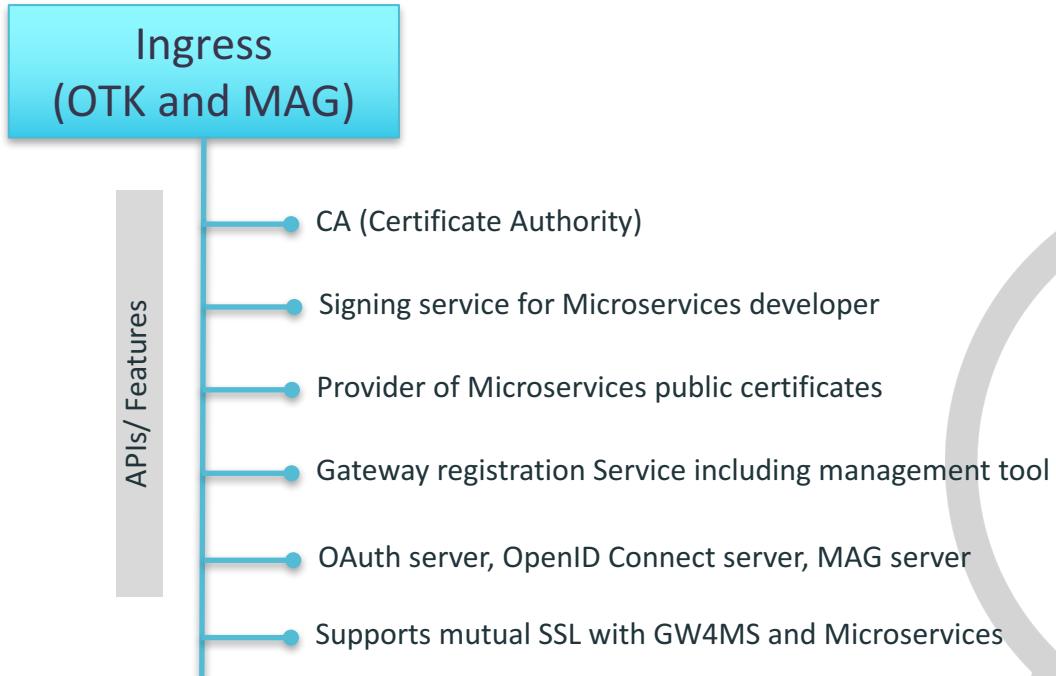
Microservices  
Gateway  
(Microgateway)

- Establish secure tunnel between Microgateway and microservices (e.g. Mutual SSL, network configuration)
- Decode JWT content into requests (e.g. HTTP headers)

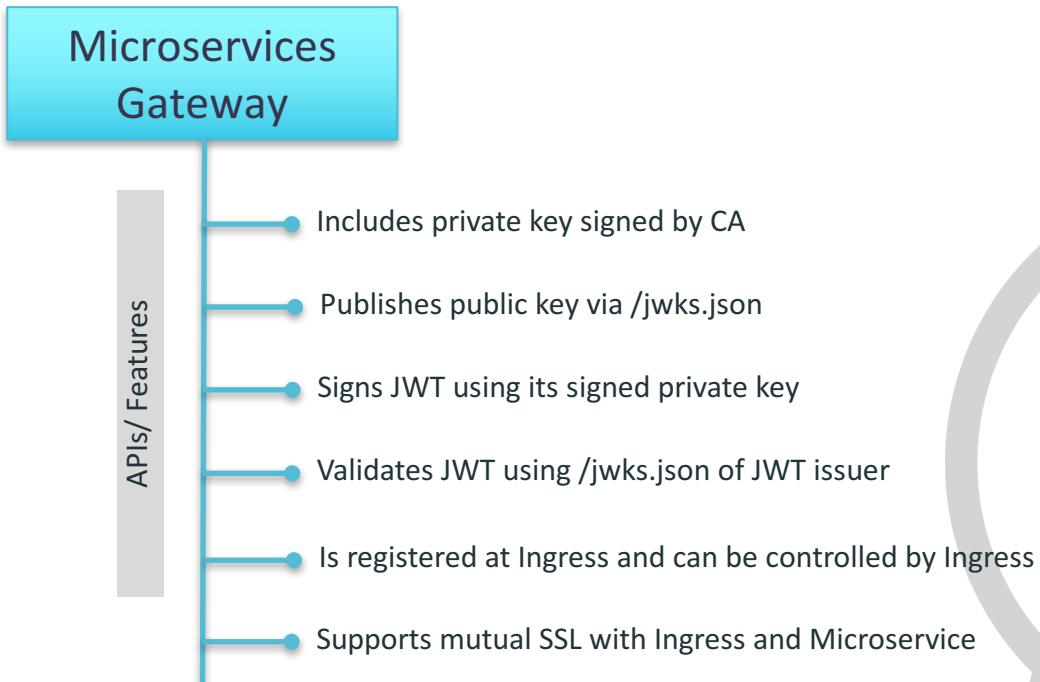
Microservice  
(customer)

- Establish secure tunnel between Microgateway and microservices (e.g. Mutual SSL, network configuration)
- Communicate with other microservices via secure APIs on Microgateway

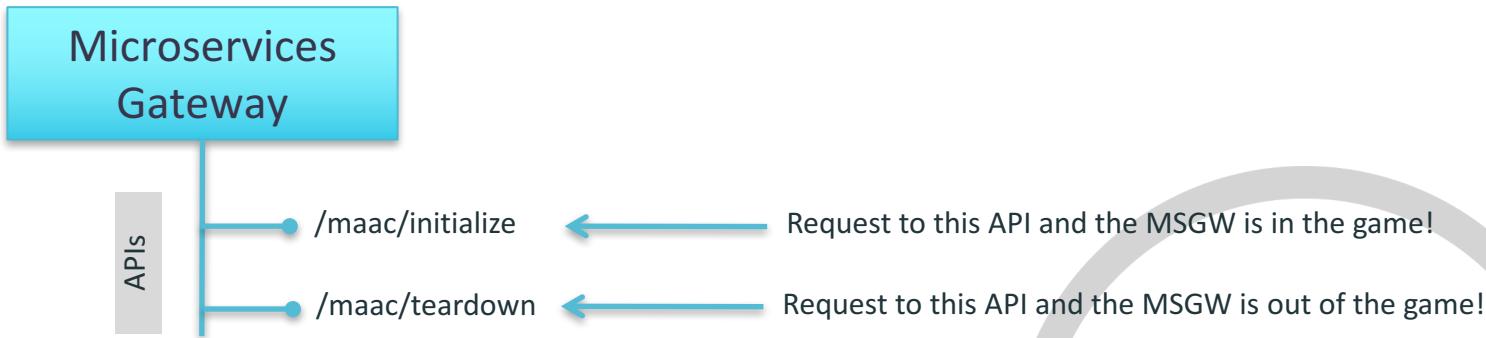
# What do we get on the Server?



# What do we get on the msgw?



# How difficult is the msgw initialization?



Initialize does the following (and teardown reverts it):

- Create private key, csr and register at Ingress
- Update listen ports to use new private key
- Update /jwks.json to publish new public key
- Register JWK (new public key) at Ingress
- Update Encode/Decode JWT assertion to use new private key

**DEMO**



## Core Concerns \*

Externalize API: How do external clients communicate with the services?

Service Discovery: How does the client of an RPI-based service discover the network location of a service instance?

Reliability: How to prevent a network or service failure from cascading to other services?

Security: How to communicate the identity of the requestor to the services that handle the request?

UI patterns: How to implement a UI screen or page that displays data from multiple services?

## Communication Patterns

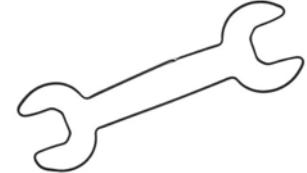
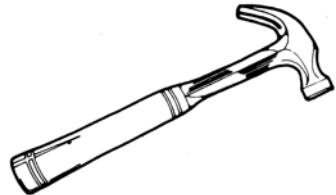
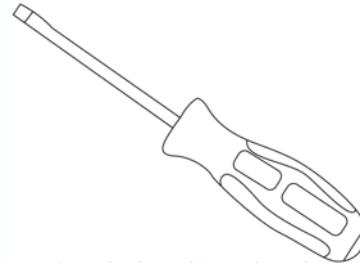
```
{"pattern": [  
    "Backend for Front End",  
    "Server-Side Discovery",  
    "Circuit-Breaker",  
    "Access-Token",  
    "Server-Side Page Composition"]}
```



CA APIM microgateway

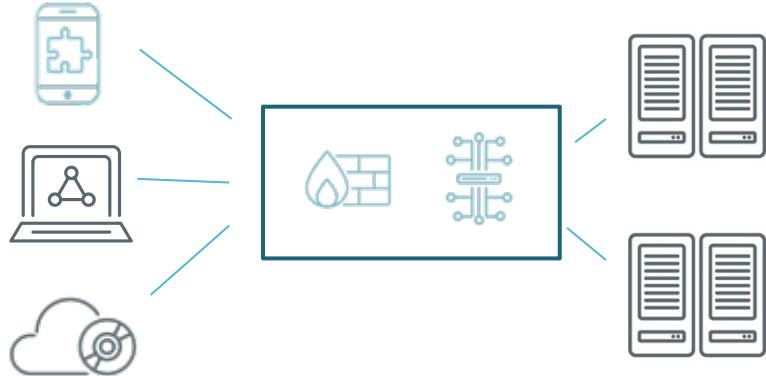
\* <http://microservices.io/patterns/index.html>

# The Role of an API Gateway In a Microservice Architecture



```
{"pattern": [ "Backend for Front End","API Gateway"] }
```

- rather than provide a one-size-fits-all style API, the API gateway can expose a different API for each client



#### **MICROSERVICE GATEWAY SECURITY FEATURES**

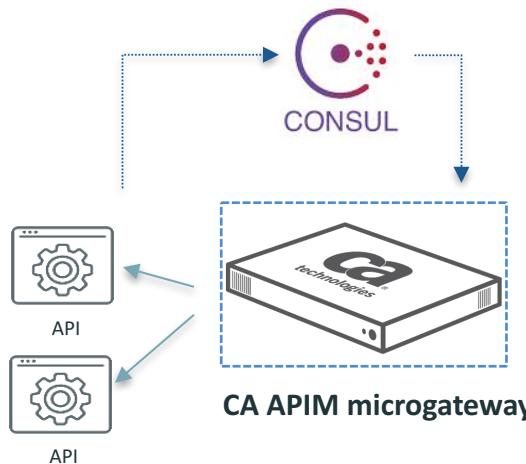
Require basic authentication credentials  
Require SSL connection  
Support CORS (cross-origin resource sharing)  
Protect again code injection attacks

#### **MICROSERVICE GATEWAY MEDIATION FEATURES**

Throttle/rate-limit API requests  
Set quota limits on API requests

# {"pattern": "Server-Side Discovery" }

- to implement a mechanism that enables the clients of service to make requests to a dynamically changing set of ephemeral service instances

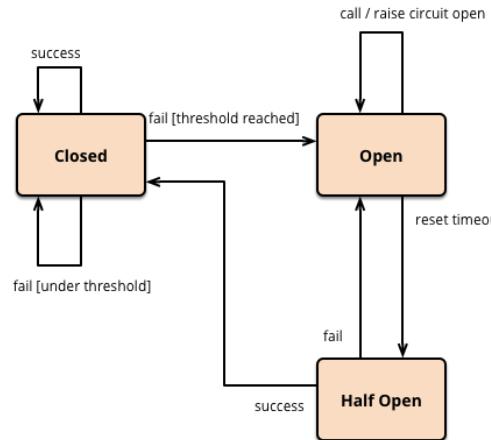


## MICROSERVICE GATEWAY SUPPORTING CAPABILITIES

- out-of-box collection of gateway functionality ("policy templates") for building meta policies including CONSUL discovery
- microgateway itself deployed as an ephemeral Docker container integrated with PaaS services which can be auto-scaled, monitored and managed with standard CI/CD tools
- future-proofed - build custom functionality (i.e., by building custom policy templates)

# {"pattern": "Circuit Breaker" }

- to prevent a network or service failure from cascading to other services



## MICROSERVICE GATEWAY MEDIATION FEATURES

Supports circuit breaker pattern including the orchestration of calls to multiple backend services and aggregate the responses

Throttle/rate-limit API requests to prevent cascading failures

<https://martinfowler.com/bliki/CircuitBreaker.html>

# {"pattern": "Access Token" }

- to communicate the identity of the requestor to the services that handle the request

The API Gateway authenticates the request and passes an access token (e.g. JSON Web Token) that securely identifies the requestor in each request to the services. A service can include the access token in requests it makes to other services.\*

## MICROSERVICE GATEWAY SECURITY FEATURES

Require basic authentication credentials  
Require SSL connection  
Require OAuth 2 token

\* <http://microservices.io/patterns/security/access-token.html>

# {"pattern": "Server-Side Page Composition" }

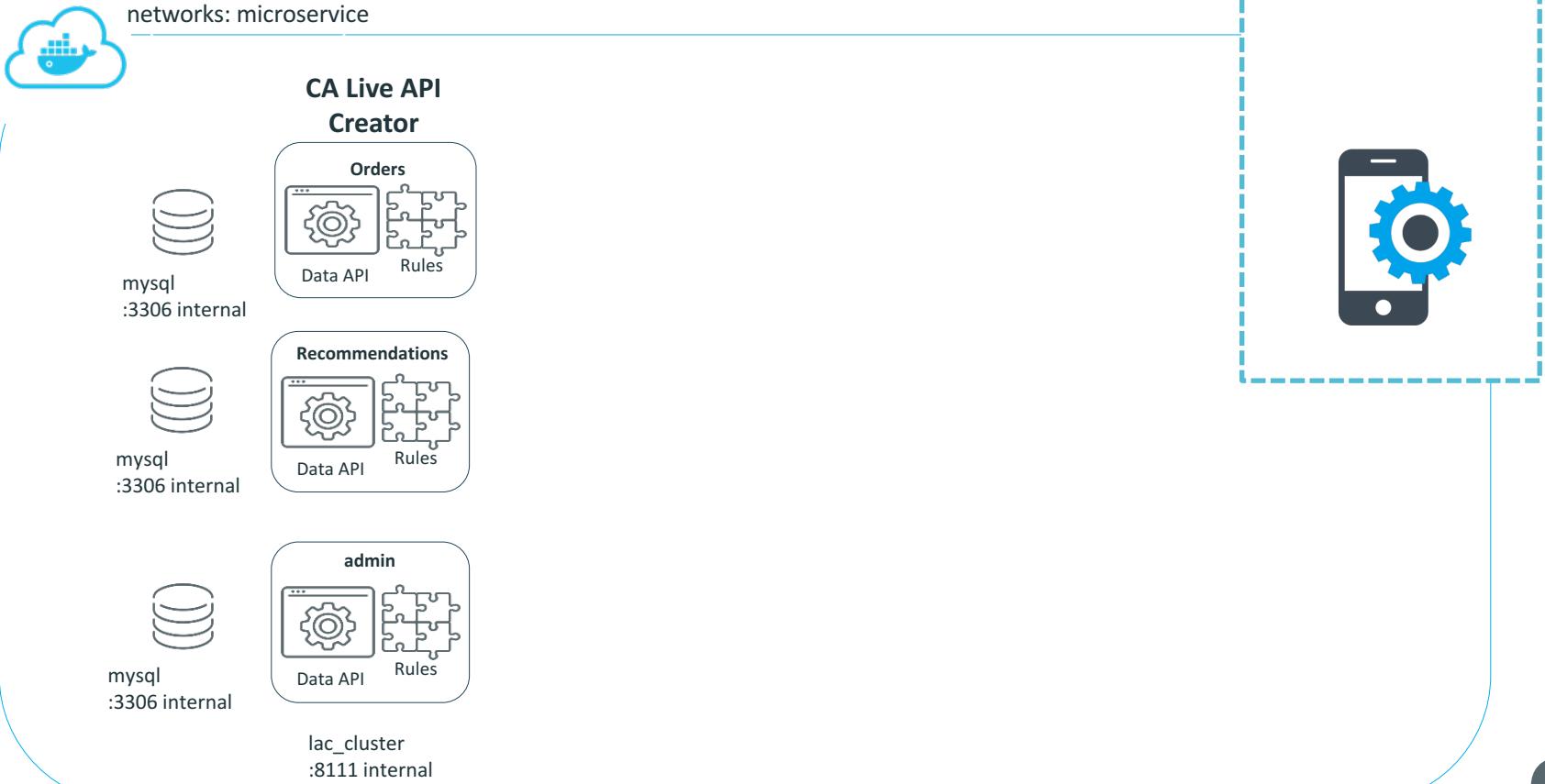
- to implement a UI screen or page that displays data from multiple services

The API Gateway orchestrates and aggregates multiple calls on behalf of the client then assembles, transforms, and offers an API that can be used for page composition

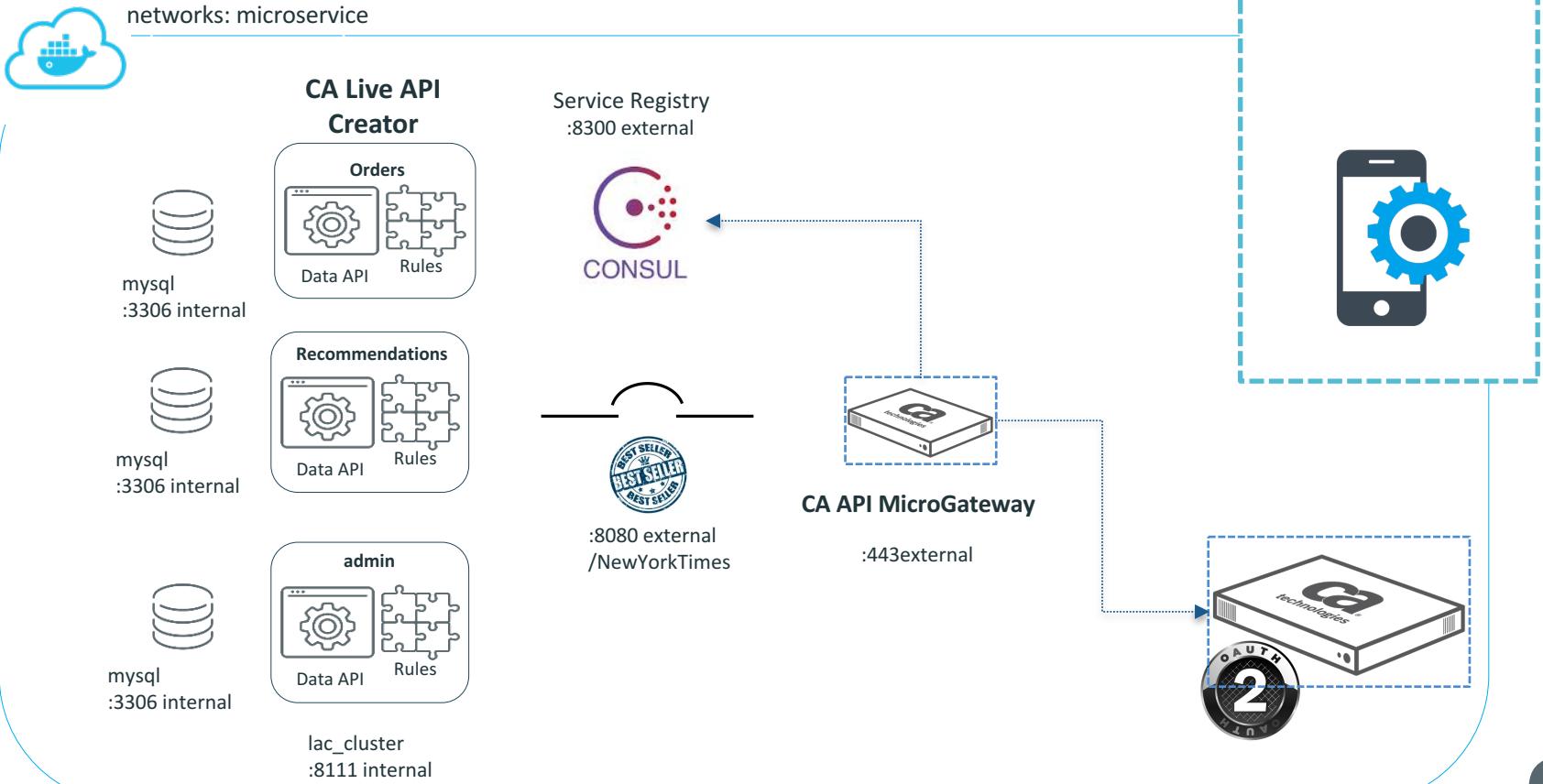
## MICROSERVICE MEDIATION FEATURES

Orchestrate calls to multiple backend services and aggregate the responses while supporting the circuit breaker pattern

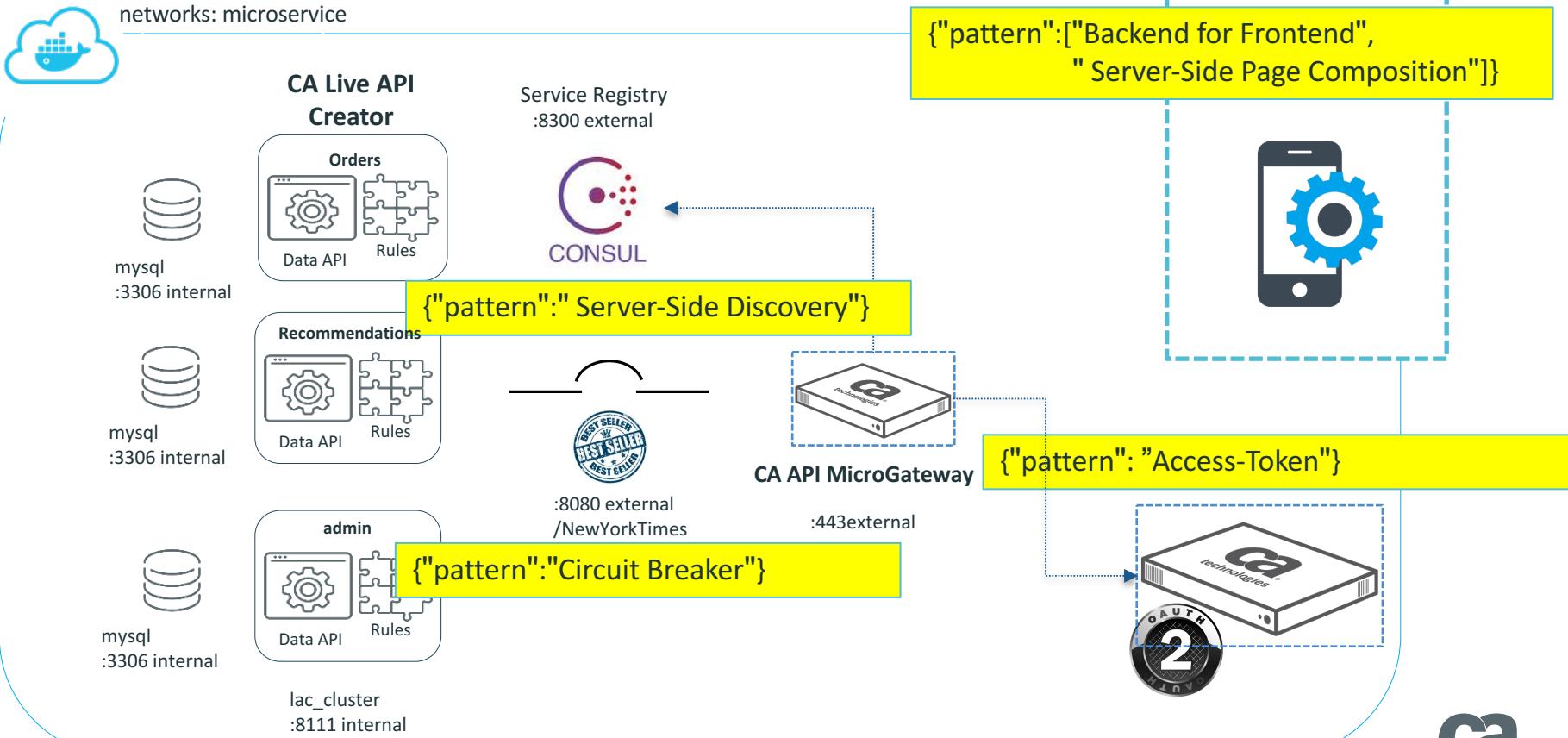
# Putting it all together – Demo Scenario



# Putting it all together – Demo Scenario



# Putting it all together – Demo Scenario



# Keeping Developers in their Environment

```
1 { "Service": {  
2     "name": "order微服务",  
3     "gatewayUri": "recSvc/v1/users/*/orders",  
4     "httpMethods": [ "get", "put" ],  
5     "policy": [  
6         {  
7             "RequireSsl": {  
8                 "sslTlsRequirement": "optional"  
9             }  
10        },  
11        {  
12            "RequireOAuth2Token": {},  
13            "scope_required": "oob",  
14            "scope_fail": "false",  
15            "onetime": "false",  
16            "given_access_token": ""  
17        }  
18    },  
19    {  
20        "Cors": {}  
21    },  
22    {  
23        "custom_consul": {  
24            "consul_API1": "http://consul:8500/v1/health/service/lac_admin",  
25            "consul_API2": "http://consul:8500/v1/health/service/orders_recommendations_microservice"  
26        }  
27    },  
28    {  
29        "custom_circuitBreaker": {  
30            "primaryRoute": "http://${recommendations.result}:8080/rest/default/svcRecs/v1/recommendations",  
31            "secondaryRoute": "http://localhost:8080/newyorktimes"  
32        }  
33    },  
34    {  
35        "custom_orderAPI": {}  
36    },  
37    {  
38        "custom_recommendationsAPI": {}  
39    },  
40    {  
41        "RateLimit": {  
42            "maxRequestsPerSecond": 250,  
43            "hardlimit": true,  
44            "counterName": "RateLimit-${request.clientId}-b0938b7ad6ff"  
45        }  
46    }  
47}  
48}  
49}
```

{"pattern": ["Backend for Frontend",  
"Server-Side Page Composition"]}

{"pattern": "Server-Side Discovery"}



CA API MicroGateway

{"pattern": "Access-Token"}

{"pattern": "Circuit Breaker"}