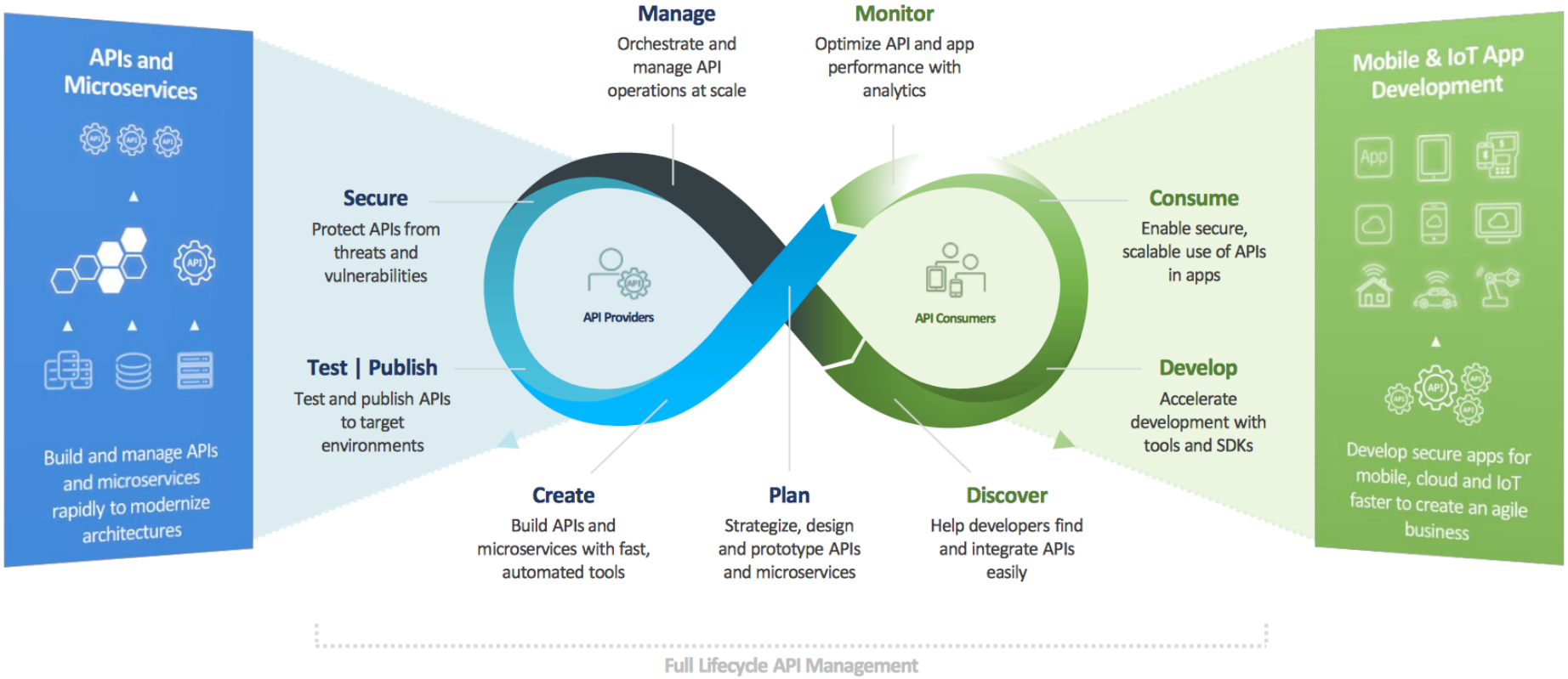
Designing and building APIs in the Microservices era

API Days Zürich 2017

Sven Walther

ca technologies

My passion: API Management



© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Agenda

1. API TRENDS - MICROSERVICES
2. EVOLUTION OF APIS
3. ORGANIZATIONAL CHALLENGES
4. DESIGNING A MICROSERVICE
5. CREATING A MICROSERVICE
6. THE BIG PICTURE

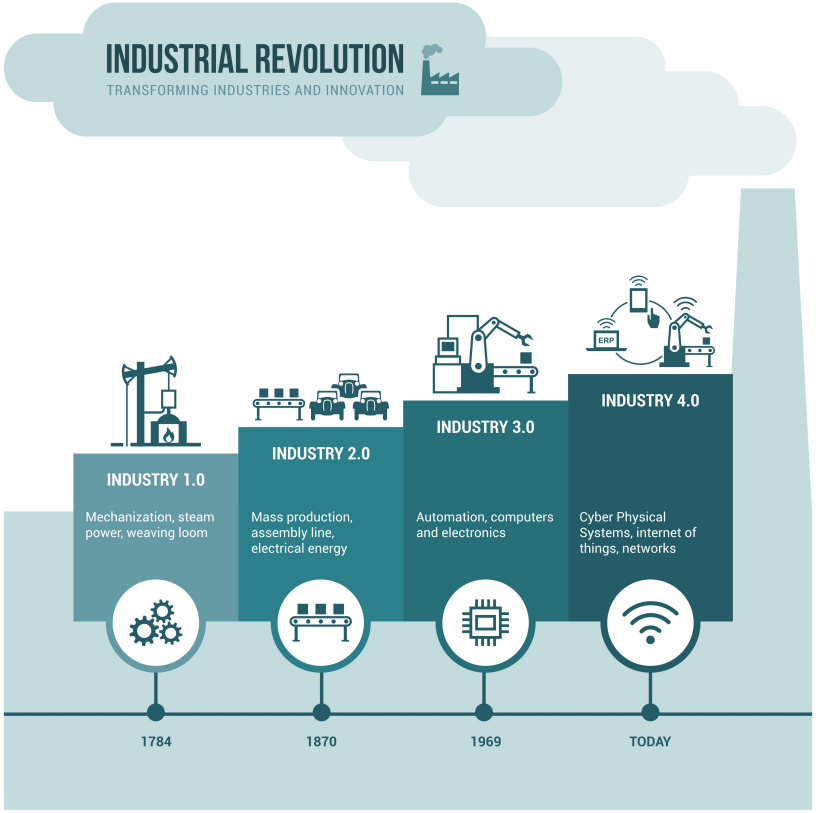
© 2017 CA. ALL RIGHTS RESERVED.

API Trends - Microservices

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

It‘s about Industry 4.0! Not 3.x!



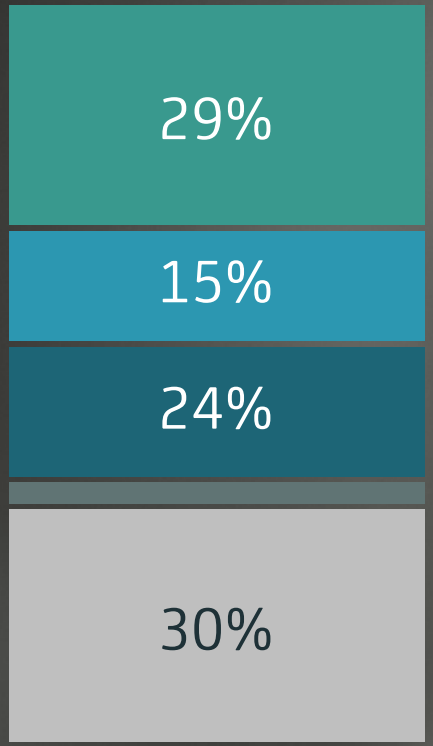
© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

SURVEY QUESTION Which statement “best” defines how your organization is currently using Microservices?

Microservices are entering mainstream

68% of organizations are using or investigating Microservices



Using Microservices in production

Using Microservices in Development

Investigating Microservices

Other (2%)

Not using Microservices

Speed and Security at Scale

Evolution of APIs

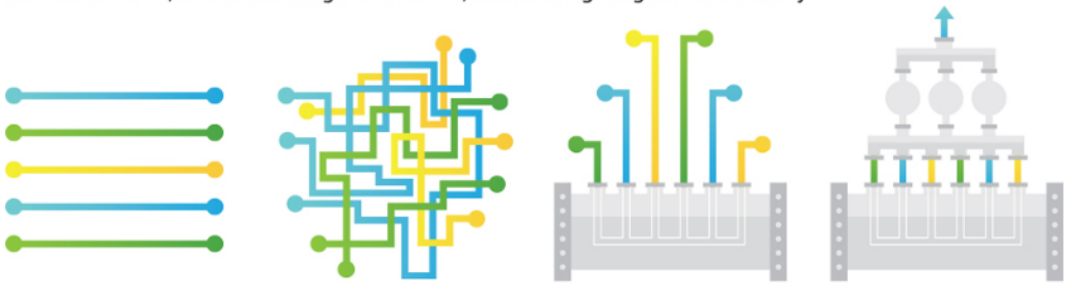
© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Deloitte. University Press

The evolution of APIs

The idea behind APIs has existed since the beginning of computing; however in the last 10 years, they have grown significantly not only in number, but also in sophistication. They are increasingly scalable, monetized, and ubiquitous, with more than 12,000 listed on ProgrammableWeb, which manages a global API directory.[a]



1960–1980

Basic interoperability enables the first programmatic exchanges of information. Simple interconnect between network protocols. Sessions established to exchange information.

TECHNIQUES

ARPANET, ATTP, and TCP sessions.

1980–1990

Creation of interfaces with function and logic. Information is shared in meaningful ways. Object brokers, procedure calls, and program calls allow remote interaction across a network.

TECHNIQUES

Point-to-point interfaces, screenscraping, RFCs, and EDI.

1990–2000

New platforms enhance exchanges through middleware. Interfaces begin to be defined as services. Tools manage the sophistication and reliability of messaging.

TECHNIQUES

Message-oriented middleware, enterprise service bus, and service-oriented architecture.

2000–today

Businesses build APIs to enable and accelerate new service development and offerings. API layers manage the OSS/BSS of integration.

TECHNIQUES

Integration as a service, RESTful services, API management, and cloud orchestration.

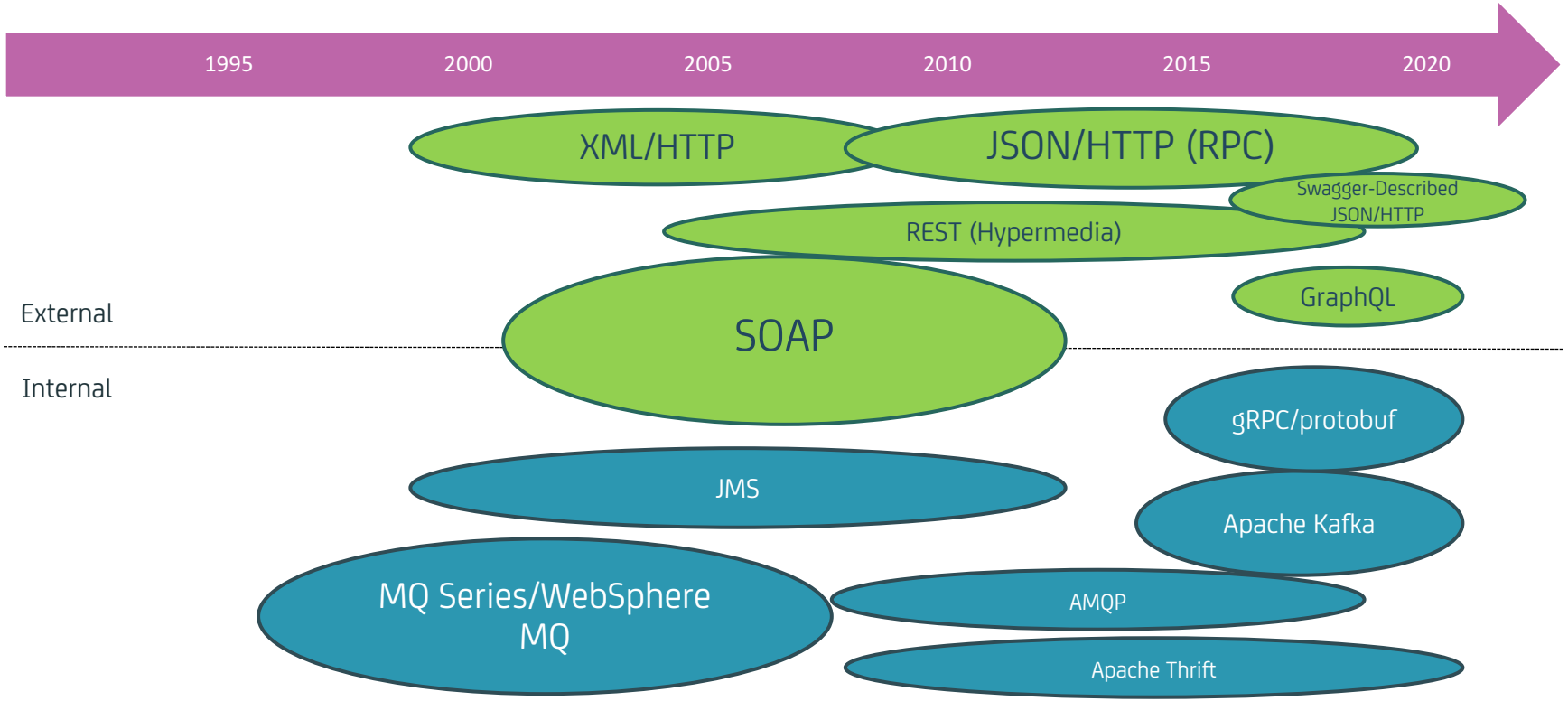
Source: [a] ProgrammableWeb, http://www.programmableweb.com, accessed January 7, 2015.

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Evolution of API Protocols

A History of Web API Protocols



© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Web API Protocols Today

- "REST" APIs are still ubiquitous
  - Mostly RPC-style JSON over HTTP
  - Increasingly described using Swagger/OpenAPI
  - Hypermedia APIs (actual REST) gaining traction
- Microservices driving adoption of new protocols
  - Event streaming (e.g. Kafka) and optimized RPC (e.g. gRPC)

© 2017 CA. ALL RIGHTS RESERVED.

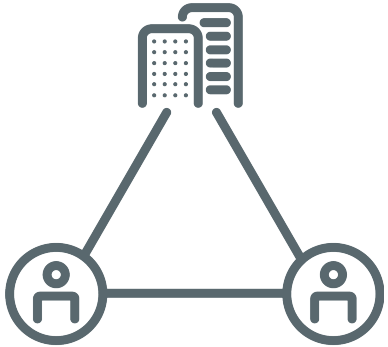
Organizational Challenges

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

How are responsibilities divided?

- Are you organized in big divisions or small teams?
- Do your teams feature all necessary roles for delivery?
  - Product Owners
  - Architects
  - Developers
  - Quality Engineers
  - Operational Engineers



© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Are your teams equipped with the right know how?

- Every teams needs to have the right skill set to deliver high quality results
- Those skills include
  - API Design
  - API Development
  - Knowledge of Distributed Applications
- If not available expect delays due to organizational processes

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

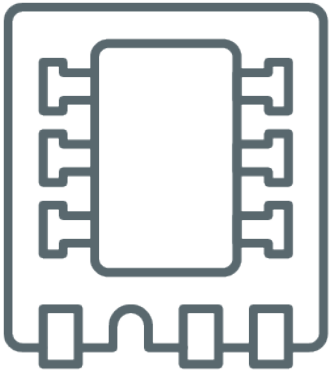
Microservices need small teams

- Create small teams fitting the need for the to be achieved goal
- To small teams will lack roles and/or knowledge
- To big teams will tend to create a incomprehensible codebase
- A team is responsible for the whole lifetime of a product so don’t treat it as a project

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Does your IT support microservices?



- Do you practice DevOps – or Dev and Opps?
- Does your deployment architecture cover the needs of Microservices?
- Are containers already being used in your infrastrucure?
- Is automatization already in place for your containers (e. g. Kubernetes?)

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Designing a Microservice

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

How micro should a microservice be?

- The answer is easy: 42[1]
- Decompose existing components into smaller subsystems
- Minimize the risk of two services sharing the same model
- Many microservice adopters have turned to Eric Evans' "domain-driven design" (DDD)[2]

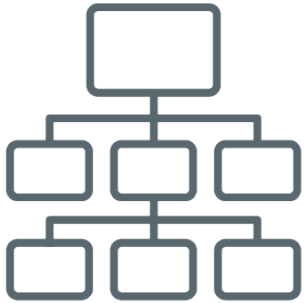
[1] The Hitchhiker's Guide to the Galaxy: https://en.wikipedia.org/wiki/42_(number)#The_Hitchhiker.27s_Guide_to_the_Galaxy
Answer to the Ultimate Question of Life, the Universe, and Everything

[2] https://en.wikipedia.org/wiki/Domain-driven_design

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Smaller is better

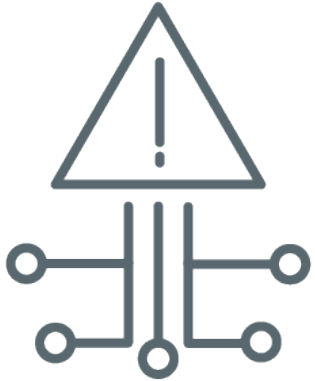


- Reduce
  - the size or scope of the problem
  - the time it takes to complete a task
  - the time it takes to get feedback
  - the size of the deployment unit
- The smaller the microservice the easier it can be developed quicker (Agile), iterated on quicker (Lean), and deployed more frequently (Continuous Delivery).
- Waterfall to Agile can be viewed as such a reduction

© 2017 CA. ALL RIGHTS RESERVED.

Smaller is not always better

- In DDD, we need a shared understanding of the domain specifics
- We cannot arbitrarily reduce the size of a bounded context because its optimal size is determined by the business context (model)



© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

API design: message oriented

- Seeing a complex system as a collection of services interchanging messages
- Depending on sender and receiver different technolgies may be applied (for example bus based internally, HTTP/JSON externally to mobile devices)

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

API design: Hypermedia

- Some companies see Hypermedia as the next level
- More than just plain data is transfered – also metadata (descriptions of possible actions) is included
- Helps the consumer to auto-discover the possible communication

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Creating a Microservice

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

A small example

- Let‘s take the classic example – an employee DB
- We need an API with typical records like
  - Name
  - Address
  - Phone
  - Email
  - Department
  - …

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Which programming language to chose?

- As the idea of a microservice is being platform and programming language independent:
- **Let the team chose**
  - But make sure they work within company standards
- Remember: You can exchange the codebase later by keeping the service description stable

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Let’s be quick

- What if we would already have the data in an encapsulated database?
- Is there a way to automate the creation of a RESTful API for it?

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Live API Creation

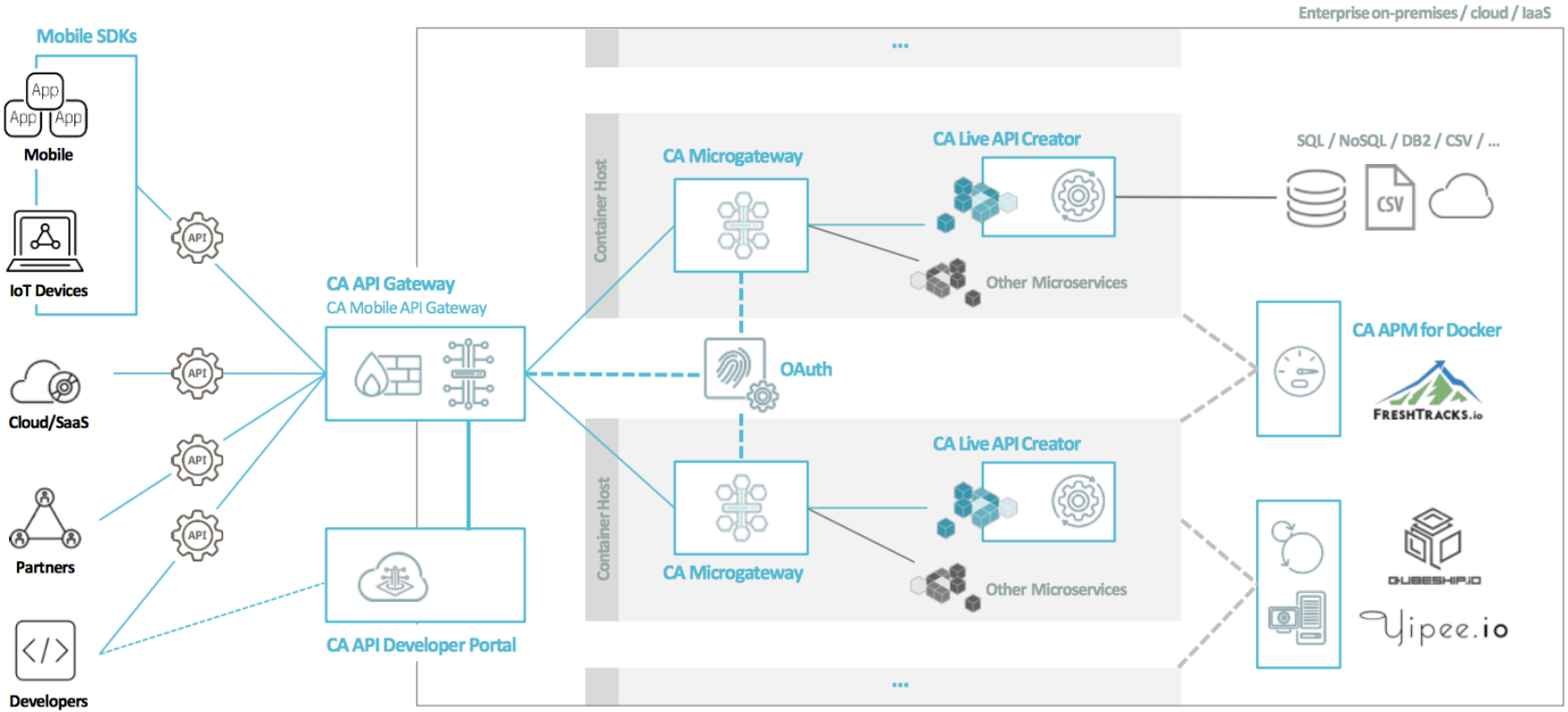
© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

The Big Picture

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies



FreshTracks.io, Qubeship.io and Yipee.io are CA Accelerator innovation projects.

© 2017 CA. ALL RIGHTS RESERVED.

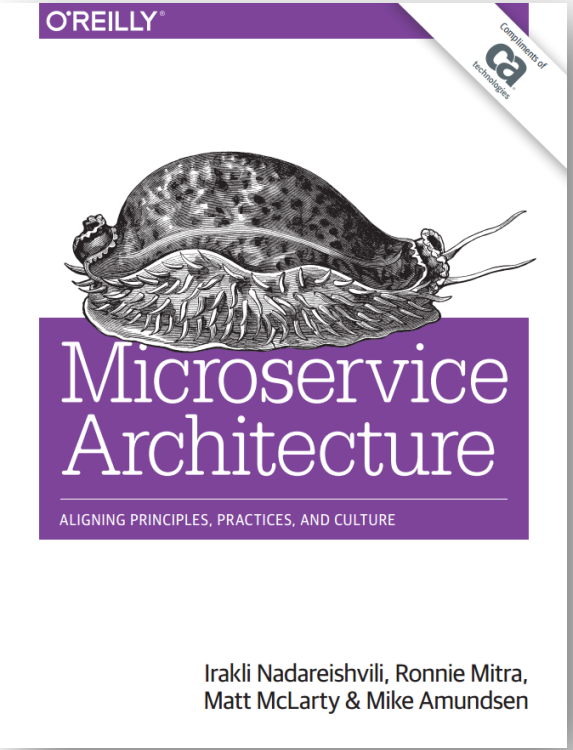
ca technologies

Links

- Documentation Live API Creator: https://docops.ca.com/ca-live-api-creator/3-2/en
- Microgateway @ GitHub: https://github.com/CAAPIM/Microgateway
- Documentation Microgateway: https://docops.ca.com/ca-microgateway/1-0/EN/

© 2017 CA. ALL RIGHTS RESERVED.

Discover more …



http://transform.ca.com/API-microservice-architecture-oreilly-book.html?source=apiacademy

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies

Thank you

sven.walther@ca.com

@SvenWal

https://www.linkedin.com/in/svenwal/

© 2017 CA. ALL RIGHTS RESERVED.

ca technologies