Serverless - introduction et perspectives concrètes

Bertrand Delacrétaz :: Principal Scientist, Adobe Basel :: Member, Apache Software Foundation :: @bdelacretaz

API DevTalks 2.0
Neuchâtel, Avril 2019

Adobe I/O

APACHE OpenWhisk

Images: Adobe Stock,
sauf indication contraire

Slides revision 2019-04-10b

#AdobeRemix
Thomas Wirtz


Adobe
Adobe Experience Manager.
Une puissante solution pour
vos systèmes CMS et DAM.
WHEREVER THE TRAILS LEADS
DISCOVER THE FASTEST GEAR
Experience
Cloud

https://www.apache.org 110%

News | About | Make a Donation | The Apache Way | Join Us | Downloads

APACHECON 2019 JOIN US !

THE APACHE SOFTWARE FOUNDATION 20TH ANNIVERSARY

SUPPORT APACHE

CELEBRATING 20 YEARS OF COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"

Projects | People | Community | License | Sponsors

THE WORLD'S LARGEST OPEN SOURCE FOUNDATION

- 200M+ lines of code in stewardship
- 1,058,321,099 lines of code committed
- 3,022,836 code commits
- 730 individual ASF Members
- 7,000 Apache code committers
- All volunteer community

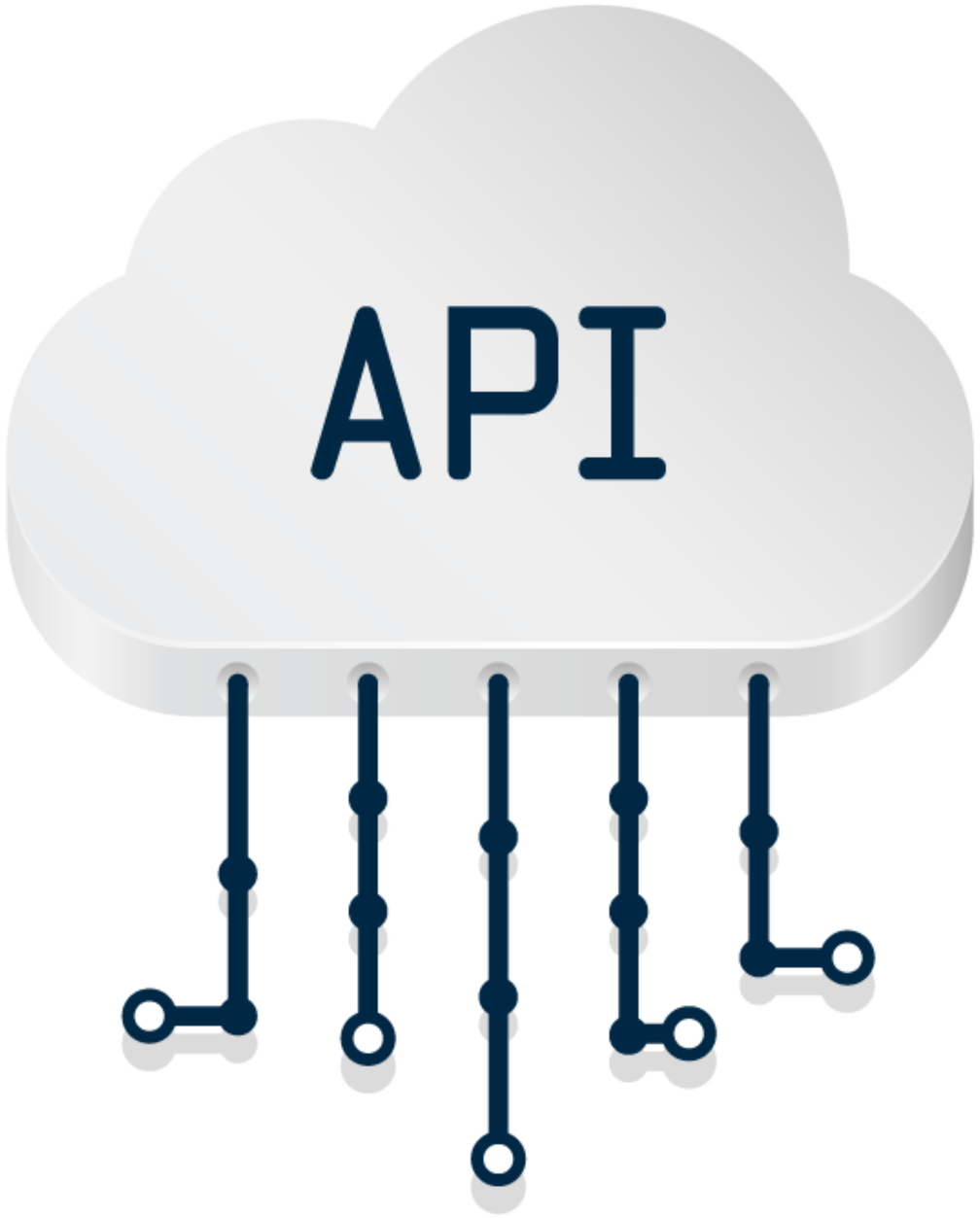
- 350+ Projects and Initiatives
- 300+ Top-Level Projects
- 52 podlings in the Apache Incubator

$20B+ worth of Apache Open Source software products are made available to the public-at-large at 100% no cost, and benefit billions of users around the world.

"The most popular open source software is Apache..."

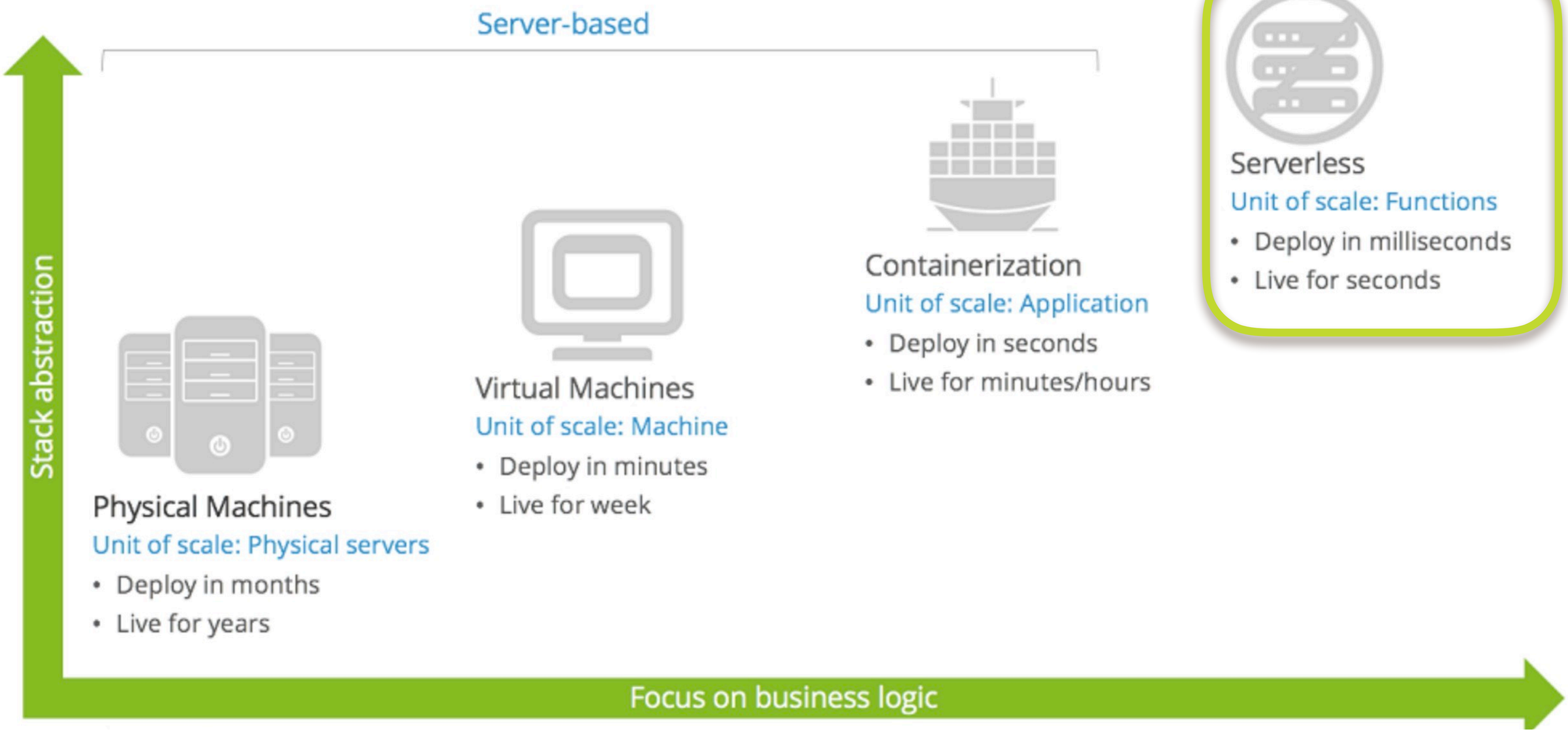
— DZone, *âWhat Open Source Software Do You Use?â*


SERVERLESS?
#AdobeRemix
Thomas Wirtz




```
$ curl https://jsonplaceholder.typicode.com/users

{
  "id": 6,
  "name": "Mrs. Dennis Schulist",
  "username": "Leopoldo_Corkery",
  "email": "Karley_Dach@jasper.info",
}
```

Evolutions des technologies dans le Cloud



Source: Deloitte Consulting LLP, via Alexander Klimetschek, @alexkli

Hello, Serverless!

```
// The actual OpenWhisk action code
function main(params) {
  const name = params.name || 'World';

  const content = `
    <html>
      <body>
        <h1>Hello, ${escapeForHTML(name)}!</h1>
      </body>
    </html>
    `;

  console.log(content);
  return {body: content };
}
```

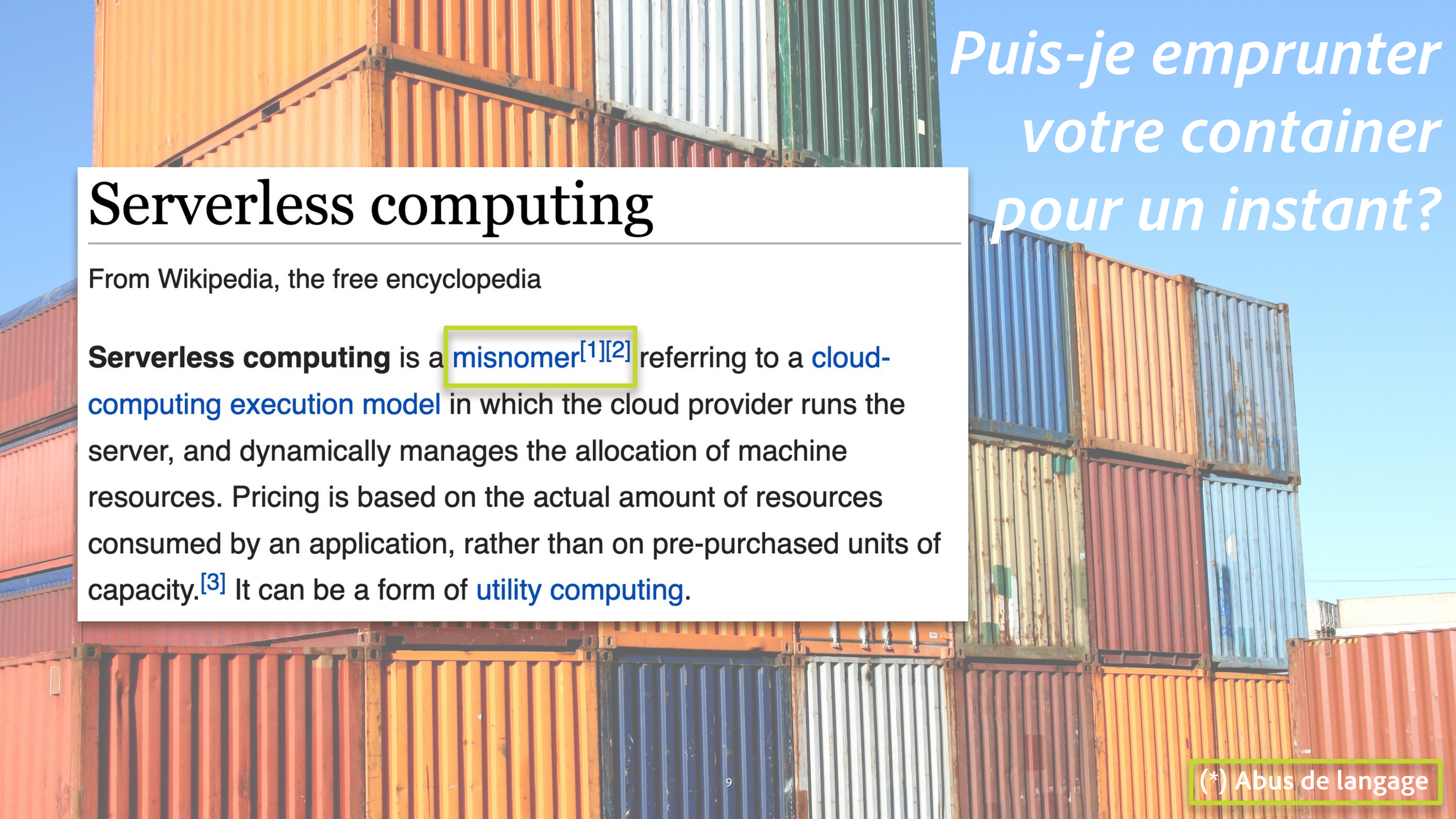
Installation:

```
wsk action update web-hello web-hello.js --web true
```

URL:

```
wsk -i action get web-hello --url
```


SERVERLESS?
#AdobeRemix
Thomas Wirtz



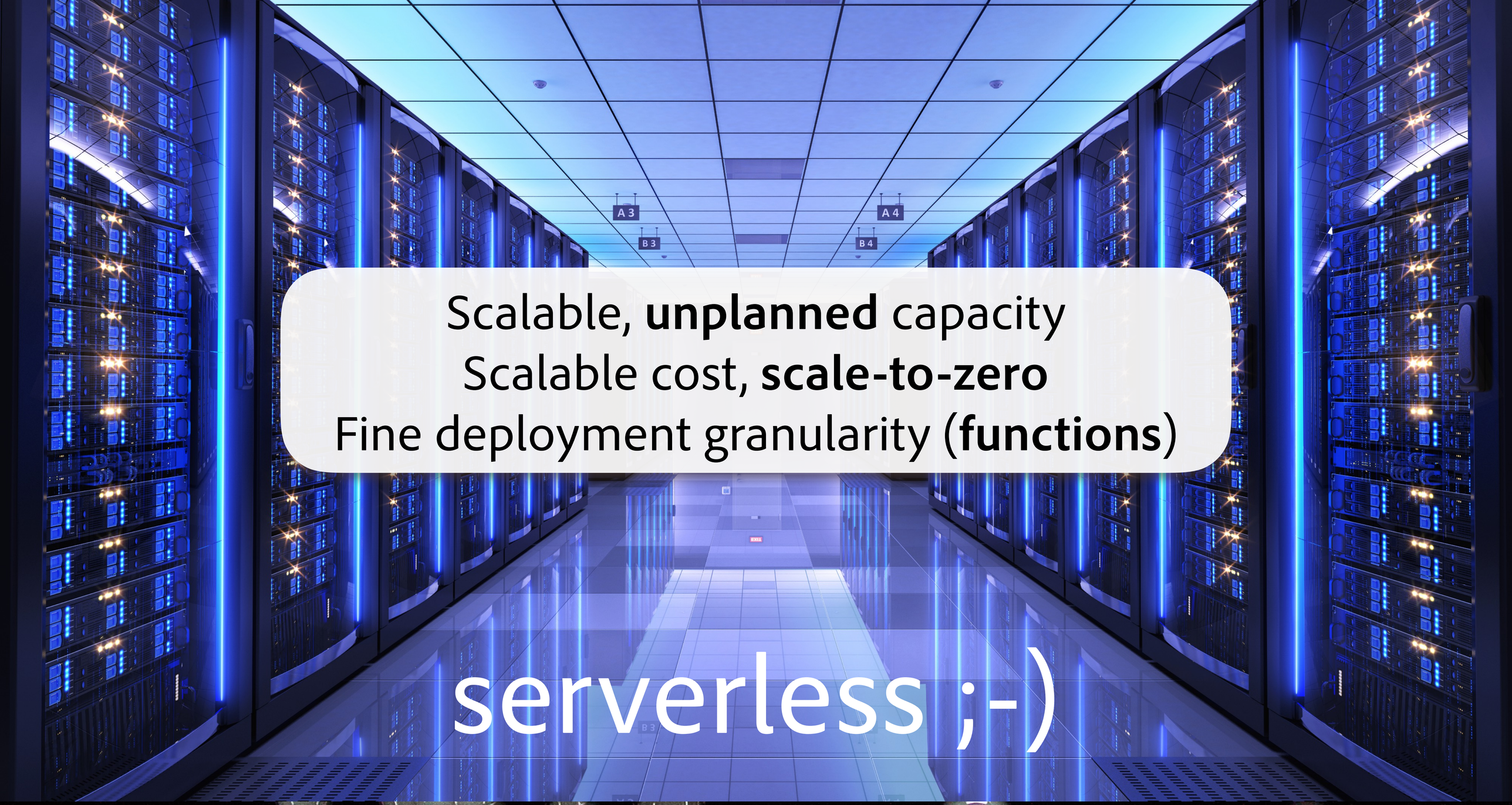
Puis-je emprunter votre container pour un instant?

Serverless computing

From Wikipedia, the free encyclopedia

Serverless computing is a misnomer[1][2] referring to a cloud-computing execution model in which the cloud provider runs the server, and dynamically manages the allocation of machine resources. Pricing is based on the actual amount of resources consumed by an application, rather than on pre-purchased units of capacity.[3] It can be a form of utility computing.

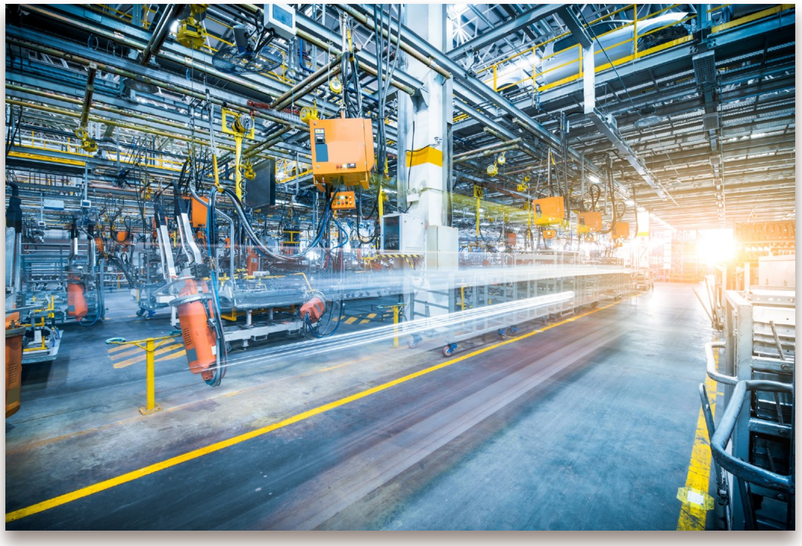
(*) Abus de langage


A 3
B 3
A 4
B 4
Scalable, **unplanned** capacity
Scalable cost, **scale-to-zero**
Fine deployment granularity (**functions**)
serverless ;-)

Une application hétérogène...

Message Bus

Factory Data



Alarms Server (Java)

Stats Server (Go)

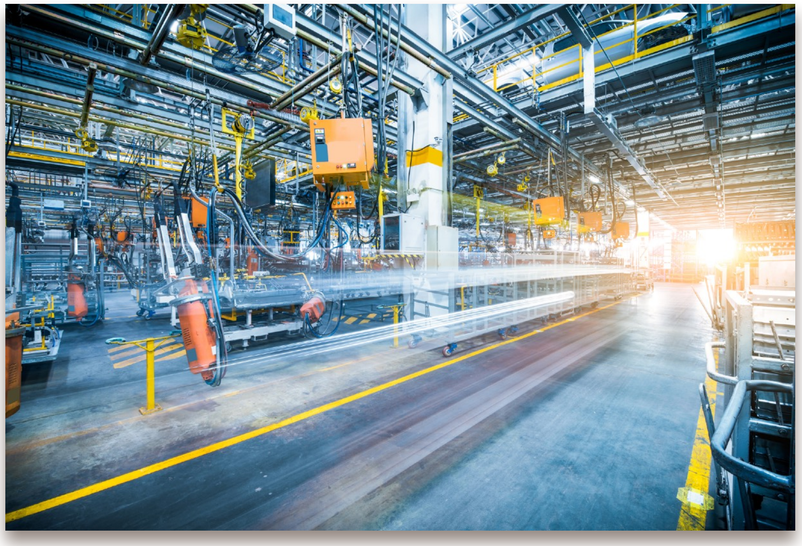
Dashboard Server (PHP)

Notifications Server (Python)

Serverless: infrastructure unique, partagée

Message Bus

Factory
Data



Alarms Functions (Java)

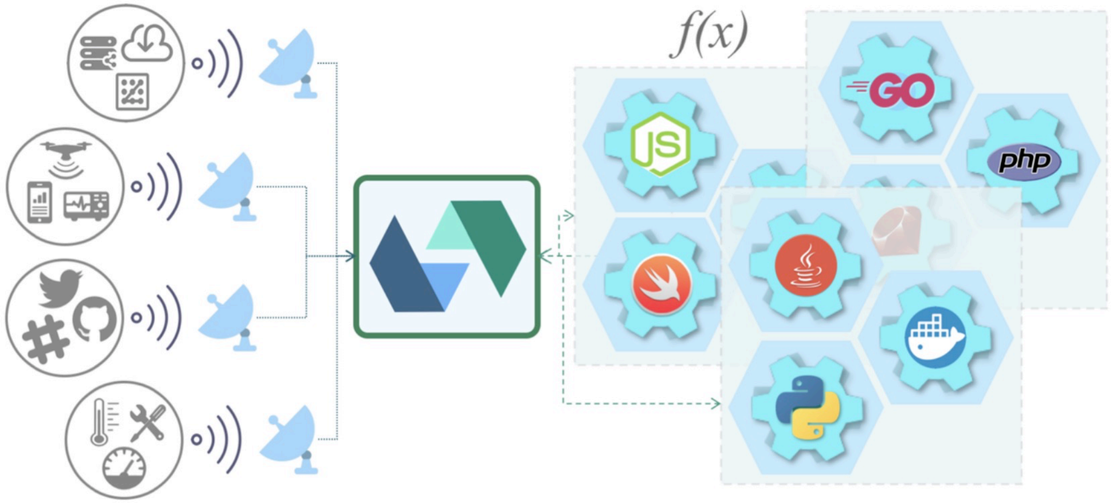
Stats Functions (Go)

Notifications Functions (Python)

Dashboard Functions (PHP)

APACHE OpenWhisk


Adobe
APACHE OPENWHISK
#AdobeRemix
Thomas Wirtz



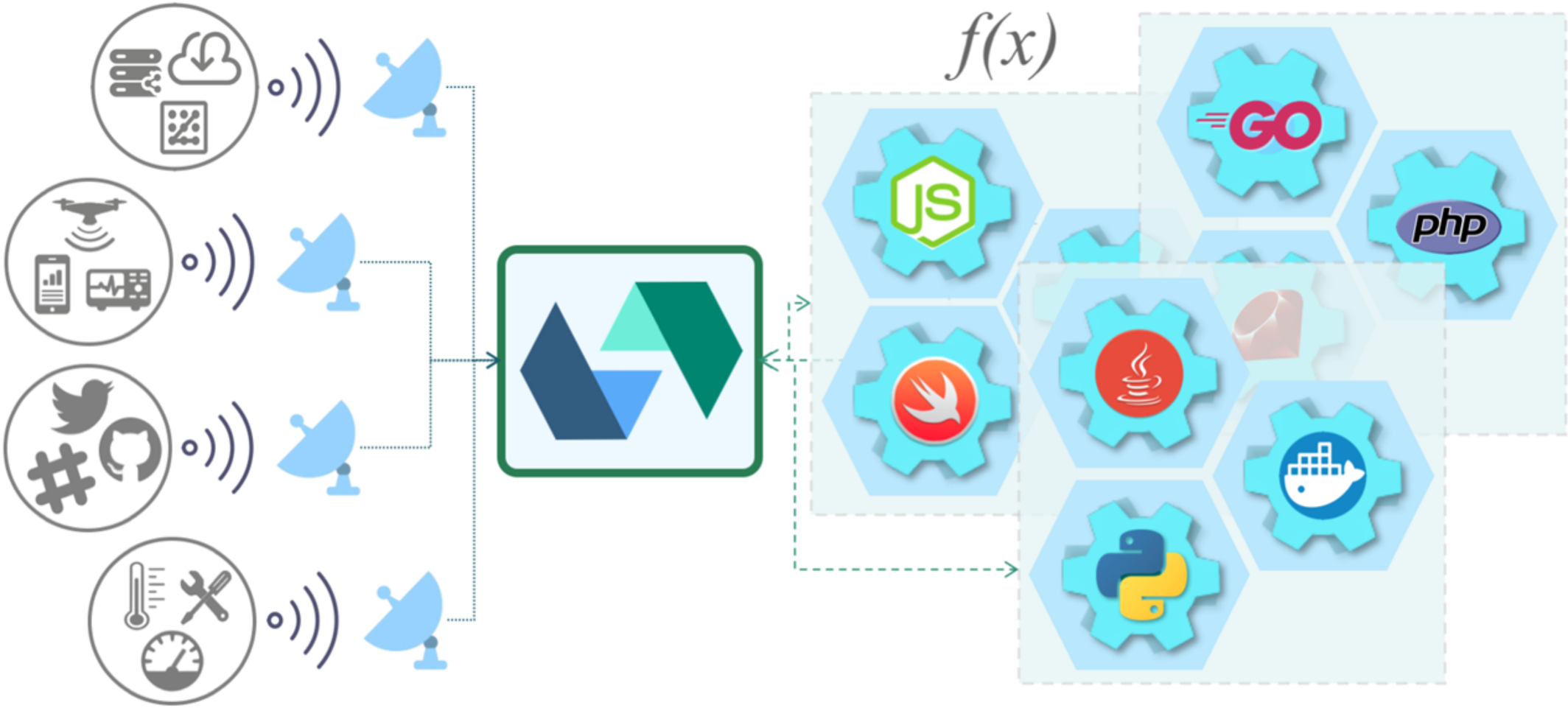
Projet en incubation de la Fondation Apache, basé sur une donation de code conjointe de IBM et Adobe, novembre 2016

http://openwhisk.incubator.apache.org

Adobe I/O runtime

Fièrement propulsé par

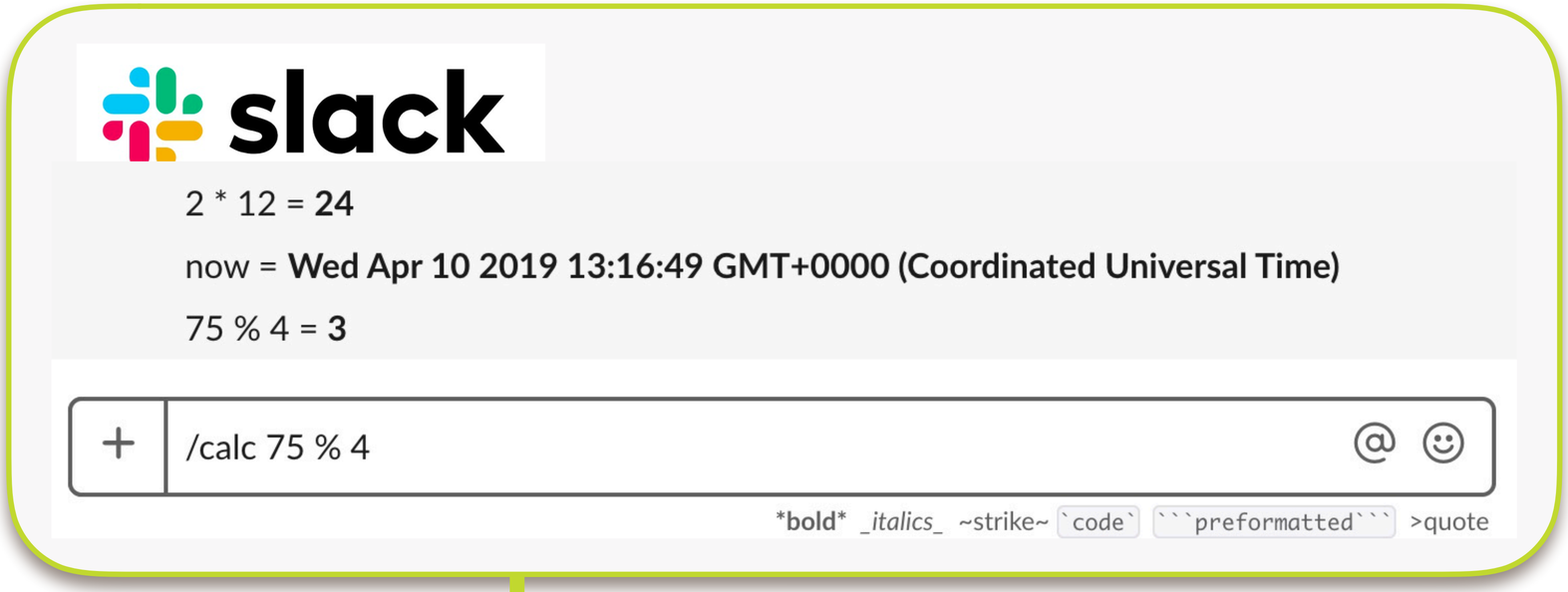
APACHE OpenWhisk™



Triggers -> OpenWhisk Core -> Orchestrated Containers

source: http://openwhisk.incubator.apache.org/

Commande Slack /calc



HTTP GET

Adobe I/O

APACHE OpenWhisk

Execute

```
async function main (params) {

  console.log(`request params: ${JSON.stringify(params, null, 2)}`)
  const expr = (params && params.text) ? params.text : "";

  try {
    return jexl.eval(expr, context)
    .then(res =>{
      const output = `${expr} = *${res}*`;
      console.log(output);
      return plainText(output);
    })
```

Commande Slack /calc

```
// The actual OpenWhisk action
async function main (params) {

  console.log(`request params: ${JSON.stringify(params, null, 2)}`)
  const expr = (params && params.text) ? params.text : "";

    try {
      return jexl.eval(expr, context)
      .then(res =>{
        const output = `${expr} = *${res}*`;
        console.log(output);
        return plainText(output);
      })
      .catch(e => {
        return plainText(`*ERROR*: ${e}`);
      })
    } catch(e) {
      return plainText(`*ERROR*: ${e}`);
    }
}
```

zip:

```
zip -r action.zip package.json *.js node_modules
```

installation:

```
wsk action update eval action.zip --web true
```

exécution:

```
export URL=$(wsk -i action get eval --url | grep http)
curl -L -k "$URL?text=2*3"
```

Slack:
Create App
Create Command (URL)

Content-addressed code!

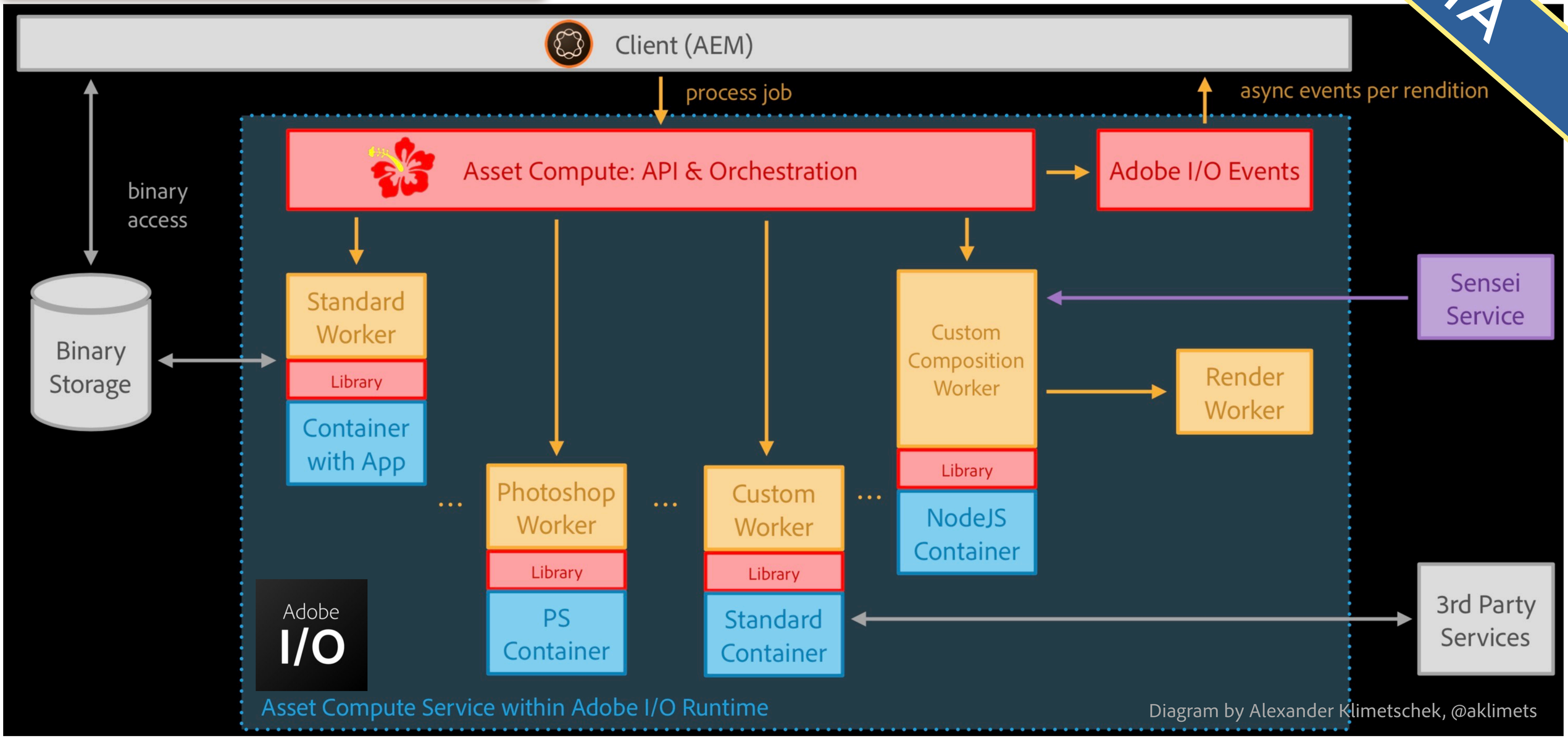
```
$ wsk action update $(md5 -q action.zip) action.zip \
> --web true --kind nodejs:10

ok: updated action 91c6b9a9d1d58ecda39e1966022ca9d9

$ wsk action get 91c6b9a9d1d58ecda39e1966022ca9d9 --url
ok: got action 91c6b9a9d1d58ecda39e1966022ca9d9

https://runtime.adobe.io/91c6b9a9d1d58ecda39e1966022ca9d9
```

Scale-to-zero: keeping multiple versions of the action code ready to run costs "nothing".


MAIS POUR QUOI FAIRE?
#AdobeRemix
Thomas Wirtz

PROJECT NUI Assets Ingestion

BETA



PROJECT NUI With Adobe Sensei

Examples by Alexander Klimetschek, @alexkli

sensei-caption-worker.js

```
// 1. ask Sensei to give us a caption for the image
return sensei(params.source, 'title-prediction')

// 2. take the highest scoring caption
.then(senseiResult => senseiResult.response[0].title.titles[0])

// 3. run imagemagick to add a caption
.then(caption => {
    console.log("CAPTION:", caption);

    // taken from http://www.imagemagick.org/Usage/annotating/
    execSync(
        `montage -label "${caption}" \
            ${source} \
            -pointsize 40 \
            -geometry +0+0 \
            -background Khaki \
            ${rendition}`
    );
});
```

deployment-package.json

```
{
  "name": "image-caption",
  "description": "Adds an automatic caption to images",
  "openwhisk": {
    "docker": "openwhisk/nodejs6action:latest",
    "memory": 4096
  },
  "dependencies": {
    "@adobe-internal-nui/library": "^0.0.6"
  }
}
```

Commerce Integration Framework

This is the cloud-based version of the Commerce Integration Framework. This framework integrates any commerce solution with the Experience Cloud, based on standardized APIs and XDM compatible data.

The integration is hosted on the serverless Adobe I/O Runtime platform and scales automatically to your needs. The modern microservice architecture supports agile development with maximal flexibility thanks to the comprehensive extensibility model.

The Commerce Integration Framework and Adobe I/O Runtime are currently in beta. Stay informed by filling out this form and we will send you an email.

https://github.com/adobe/commerce-cif-api

PASSED codecov 96%

Commerce Integration Framework (CIF) on Cloud for Magento

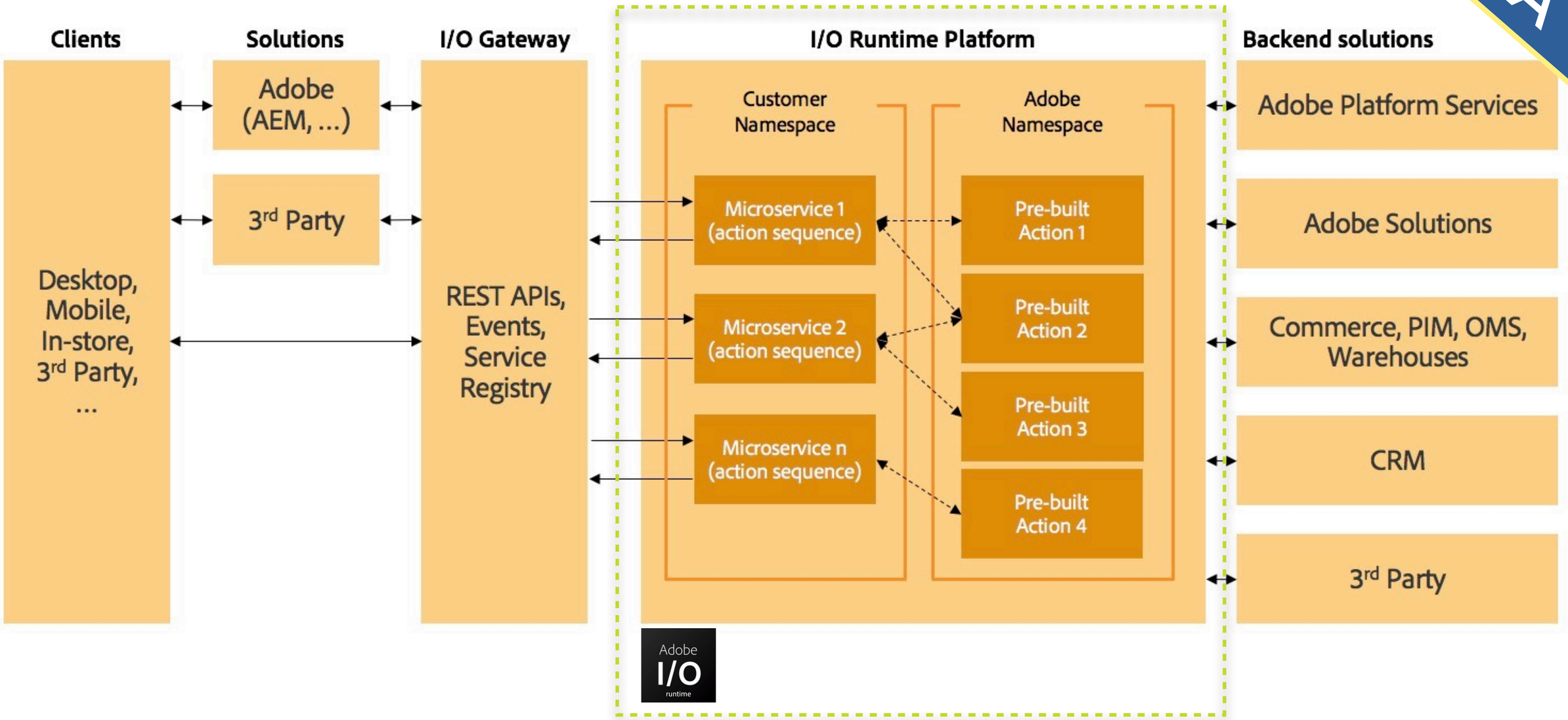
Introduction

The CIF on Cloud services architecture is based on Apache OpenWhisk & Adobe I/O Runtime. The main building blocks of the new commerce services are serverless functions (OpenWhisk actions). These actions run on Adobe I/O Runtime inside an isolated container, stateless and serverless interacting with the commerce backend system or other endpoints via their APIs.

This project contains the OpenWhisk actions implementation for Magento.

Commerce Integration Framework Architecture

BETA



source: https://www.adobe.io/apis/experiencecloud/commerce-integration-framework/docs.html

The Cloudflare Global Anycast Network

source: https://www.cloudflare.com/network/

What Worker Code Looks Like

A worker can return a static response:

```
addEventListener('fetch', event => {
  event.respondWith(new Response('hello world'))
})
```

It can also make a request to another location, and return that. In this case we're adding '/index.html' to the end of the URL:

```
addEventListener('fetch', event => {
  event.respondWith(
    fetch(event.request.url + "/index.html", event.request)
  )
})
```

Being JavaScript, it's possible to leverage these capabilities in almost any way imaginable. You could, for example, make decisions about where to route the request based on its contents:

```
addEventListener('fetch', event => {
  let url = new URL(event.request.url)

  if (event.request.headers.has('X-Use-Dev'))
    url.host = "dev." + url.host

  url.protocol = 'https:'

  event.respondWith(
    fetch(url, event.request)
  )
})
```

Edge Computing - CDN - Workers

Liens et Coda

Adobe I/O Runtime
https://www.adobe.io/apis/experienceplatform/runtime.html

Apache OpenWhisk + mes exemples
http://openwhisk.incubator.apache.org/
https://github.com/bdelacretaz/openwhisk-playground

Serverless to the Max:case study, Troy Hunt, haveibeenpwned.com
http://bit.ly/serverless-max

Developing Serverless Applications
short book by Raymond Camden, O'Reilly

Twitter:
@adobeio, @alexkli, @bdelacretaz

BETA

Adobe I/O

APACHE OpenWhisk™

Serverless -> Intégration !

Many thanks to the I/O Runtime & Friends "gang" for your contributions, you know who you are!

Adobe