

# Hálózatba kapcsolt erőforrás platformok és alkalmazásaik

Simon Csaba

TMIT

2018

# What is a function?



# Cloud Abstractions

Functions

Serverless / FaaS  
platform

Apps

Cloud Foundry

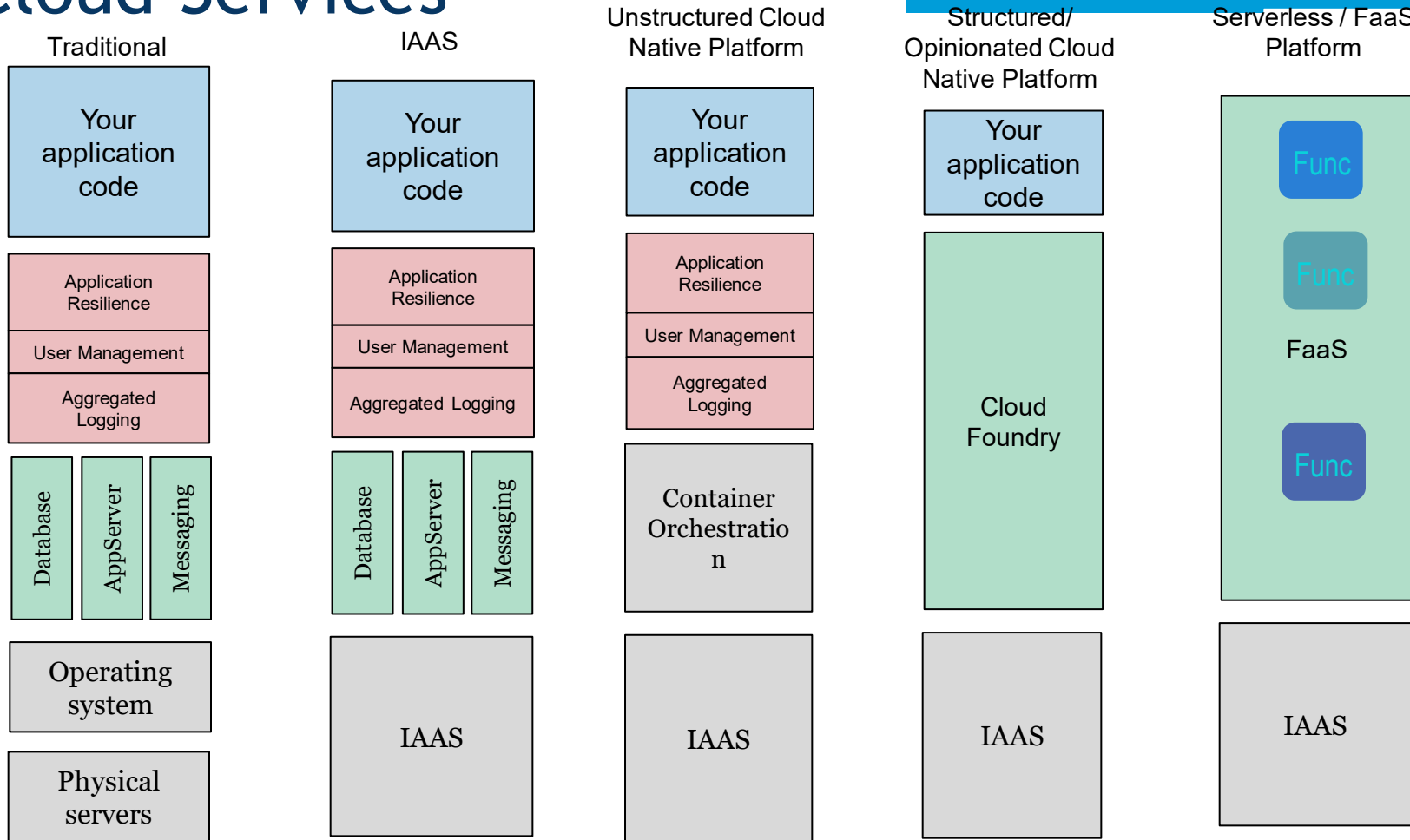
Containers

Container  
orchestration tools  
such as Kubernetes

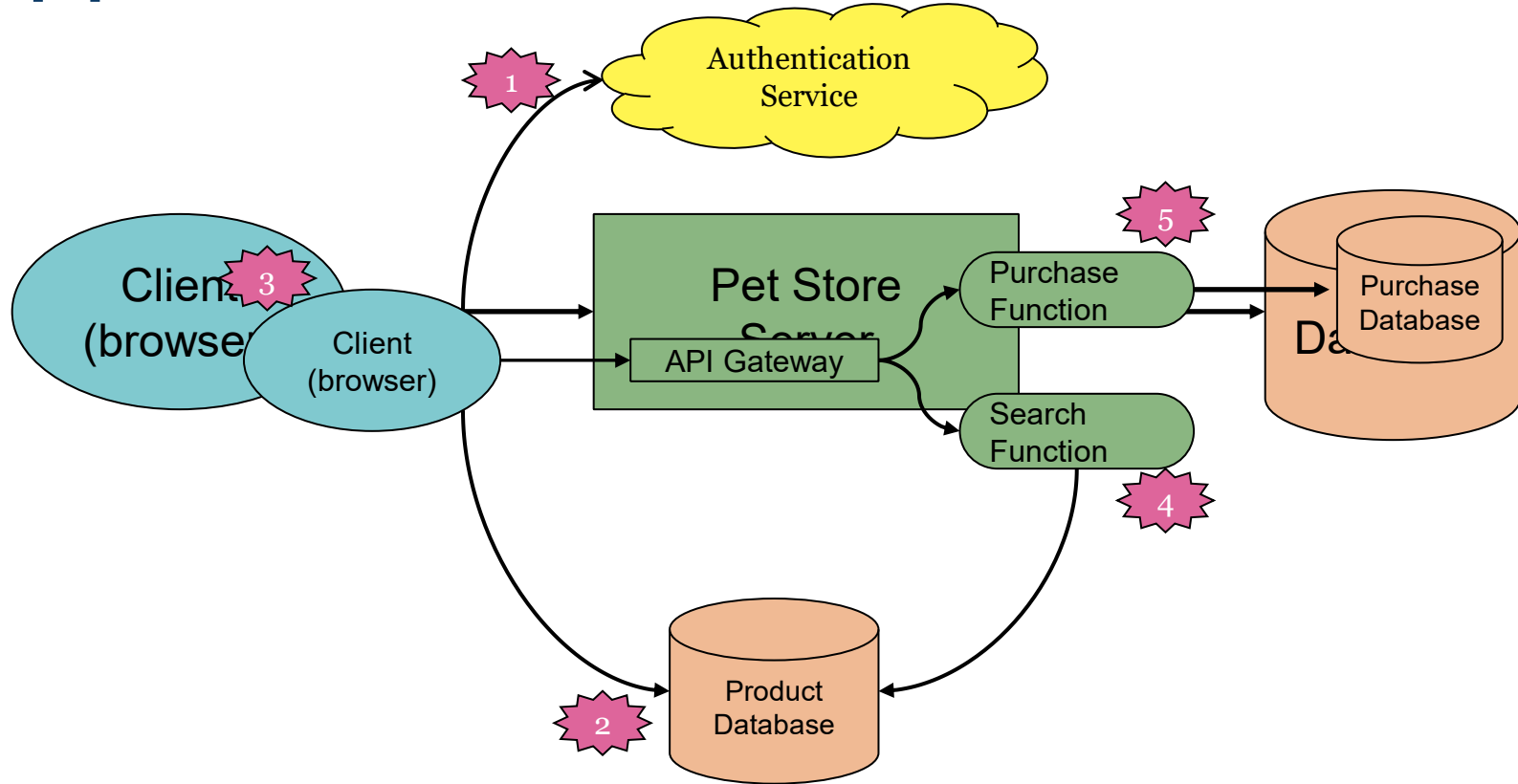
Virtual Machines

Bare Metal Servers

# Cloud Services



# Applications -> Functions



# Characteristics of Serverless / FaaS

- Focus on business logic
- Event driven
- Short lived
- Stateless
- Auto scaling and auto deployment
- Billing per execution
- Prototypes become production code really quickly

# Amazon Lambda

Lambda Management console

https://us-east-2.console.aws.amazon.com/lambda/home?region=us-east-2#/create/configure-function?firstrun=true

Services ▾ Resource Groups ▾

Planned Dave Syer ▾ Ohio ▾ Support ▾

Lambda > New function

Select blueprint

Configure triggers

Configure function

Review

Configure function

A Lambda function consists of the custom code you want to execute. [Learn more](#) about Lambda functions.

Name\*

myFunctionName

Description

Runtime\*

Java 8 ▾

Lambda function code

Provide the code for your function. [Learn more](#) about deploying Lambda functions.

Code entry type

Upload a .ZIP or JAR file ▾

Function package\*

Upload

For files larger than 10 MB, consider uploading via S3.

You can define Environment Variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#). For storing sensitive information, we recommend encrypting values using KMS and the console's encryption helpers.

Enable encryption helpers

☐

# Google Cloud Function

The screenshot shows the Google Cloud Platform console interface. At the top, there's a navigation bar with the Google Cloud Platform logo and the project name 'CF-Spring-funkytown'. Below this, the 'Cloud Functions' section is active, displaying the 'Function details' for 'helloWorldJava'. The function is in a 'Ready' state. The 'Code preview' tab is selected, showing the source code for 'Hello.java'. The code is a simple Java class with a 'main' method that prints 'Hello ' followed by the first argument. Below the code preview, the 'Source location' is listed as 'cfsf-hello-world-java/us-central1-helloWorldJava-qpllmznutury.zip'. The 'Function to execute' is also listed as 'helloWorldJava'.

Cloud Functions - CF x

https://console.cloud.google.com/functions/details/us-central1/helloWorldJava?project=cf-spring-funkytown&tab=source&duration=PT1H

Sign up for a free trial and you'll get \$300 in credit and 12 months to explore all of Google Cloud Platform. [Learn more](#) Contact sales: 0800 026 1212 (9am - 5pm GMT) DISMISS SIGN UP FOR FREE TRIAL

Google Cloud Platform CF-Spring-funkytown

Cloud Functions Function details EDIT DELETE COPY VIEW LOGS

helloWorldJava

General Trigger Source Testing

Code preview

Hello.class Hello.java index.js package.json

```
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello " + args[0]);
    }
}
```

Source location

[cfsf-hello-world-java/us-central1-helloWorldJava-qpllmznutury.zip](#)

Function to execute

helloWorldJava



# Serverless Providers

- (J) Amazon Lambda
- Google Cloud Functions
- Azure Function
- (J) Apache OpenWhisk
- Fission
- Kubeless
- ...

(J) = native Java support

Others can run Java via node JRE launcher.

Multiple language support:

- NodeJS
- Python
- Java
- Scala
- Clojure
- ...

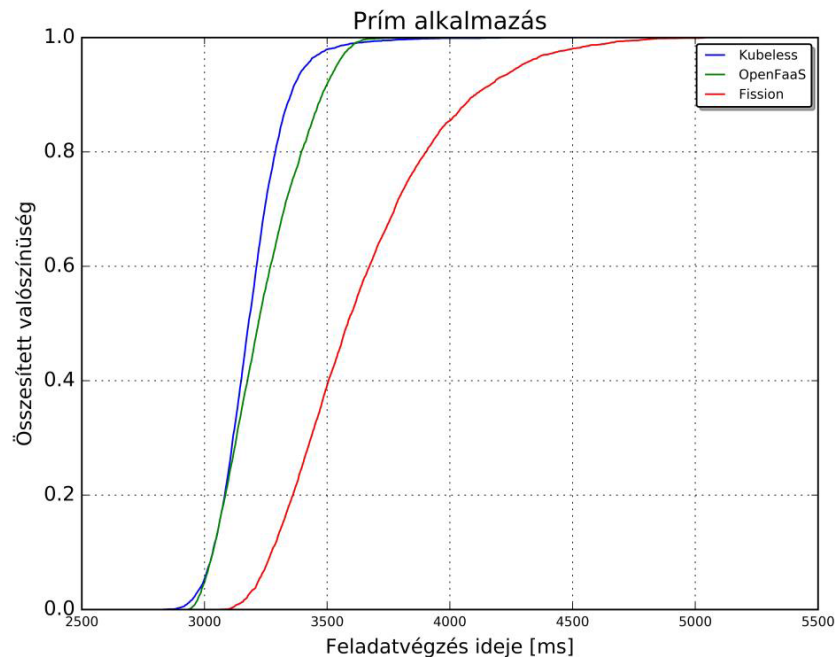
# Problems/concerns

- **New architectural style.** The serverless style of truly stateless, mostly asynchronous, horizontally scaled apps takes 12-factor to the extreme. It's a positive trend, but one that is not natural for many developers today.
- **IaaS lock-in.** Serverless platforms today are cloud only, and not portable between vendors. Also, while the serverless code itself is portable, the unique connectors/services/triggers in each cloud make migration a challenge.
- **Immature tooling for development.** Substandard tools available today for IDE integration, tests, continuous integration.
- **Immature tooling for “day 2.”** As expected for new tools and new paradigms, the support tooling is fairly sparse. Monitoring, logging, debugging, tracing, and versioning tools are fairly weak.

# Challenges of Serverless / FaaS

- New architectural style
- Management of large populations of functions
- Vendor lock-in
- Execution duration limit
- Start up latency
- Network latency among functions
- Immature tooling for development
- Immature tooling for Day 2

# The problem of FaaS: eventually high response times



# Gartner's case studies

## Most Cited Advantages

- Cost-efficiency
- Simplicity

## Most Cited Disadvantages

- Lack of tooling
- Learning a new programming model

## Most Cited Lessons Learned

- Begin adoption sooner rather than later
- It can truly bring transformative change

Is serverless a real game changer or just the latest fad?

## Gartner's answer

***“The years 2017 and 2018 should reveal whether serverless computing will fall into the Trough of Disillusionment to be counted as just another overhyped alternative, or if it emerges quickly as a real step forward in cloud application architecture.”***

# Gartner's strategic planning assumption

***“By 2022, over 50% of cloud-based software offerings will incorporate some form of serverless technology (function platform as a service [fPaaS], database as a service [DBaaS], object storage, and so on), up from less than 10% in 2017.”***

# Next Steps

- More toolings, patterns and platform improvements
- Mindset changes on software design
- Validations and justifications to adopt serverless technology



# Links

- Spring Cloud Function: <https://github.com/spring-cloud/spring-cloud-function>
- Fission: <http://fission.io/>
- 12 factor: <https://12factor.net/>
- Spring Initializr: <http://start.spring.io>
- Spring Cloud: <http://cloud.spring.io>
- Reactor: <http://projectreactor.io>

# Kubeless

- [kubeless.io](https://kubeless.io)
- <https://github.com/kubeless/kubeless>

Open Source , non-affiliated

- <https://github.com/kubeless>



- Kubernetes extension
- TPR (CRD) for functions
- A controller
- Creates deployments, services
- Creates Ingress if needed
- Uses Configmap, Init container for now
- Instrumented runtimes with prometheus-client
- A cute UI
- A serverless Plugin
- Custom metrics HPA coming ...

# Application types

- Event based distributed apps
- Composed of many different services triggered by events
- AWS Lambda Clone
- Google Cloud functions like CLI
- Same realm as OpenWhisk, Fission ...

# Serverless deployment framework



Today in Serverless 1.20 we now have kubeless support:

20k stars, the *go to* framework to deploy/managed serverless functions on lambda, Azure, GCF, OpenWhisk.

```
sls create --template kubeless-python --path mypythonfunction
```

```
sls create --template kubeless-nodejs --path mynodefunction
```

- <https://serverless.com>

# Scripting the fn deployment process

Open handler.py and add this function

```
def test():  
    list_element = [1,5,8,12,78,12]  
    for i in range(len(list_element)):  
        print list_element[i]
```



```
service: test  
  
provider:  
  name: kubeless  
  runtime: python2.7  
  
plugins:  
  - serverless-kubeless  
  
(The above one is the plugin name as I am  
using serverless-kubeless plugin)  
  
functions:  
  test:  
    description: 'print the array'  
    handler: handler.test
```



serverless deploy