CLOUD
DEVELOPER
DAYS

Todor Todorov | @totollygeek

Cloud Agnostic Serverless with Fn Project

STRATEGIC SPONSORS

GOLD SPONSORS

CLOUD DEVELOPER DAYS

Todor Todorov

Senior Software Engineer

- ✓ .NET developer;
- ✓ clean code fanatic;
- ✓ DevOps evangelist;
- ✓ speaker;
- ✓ father of 3 boys;
- ✓ karaoke enthusiast;

The serverless explained

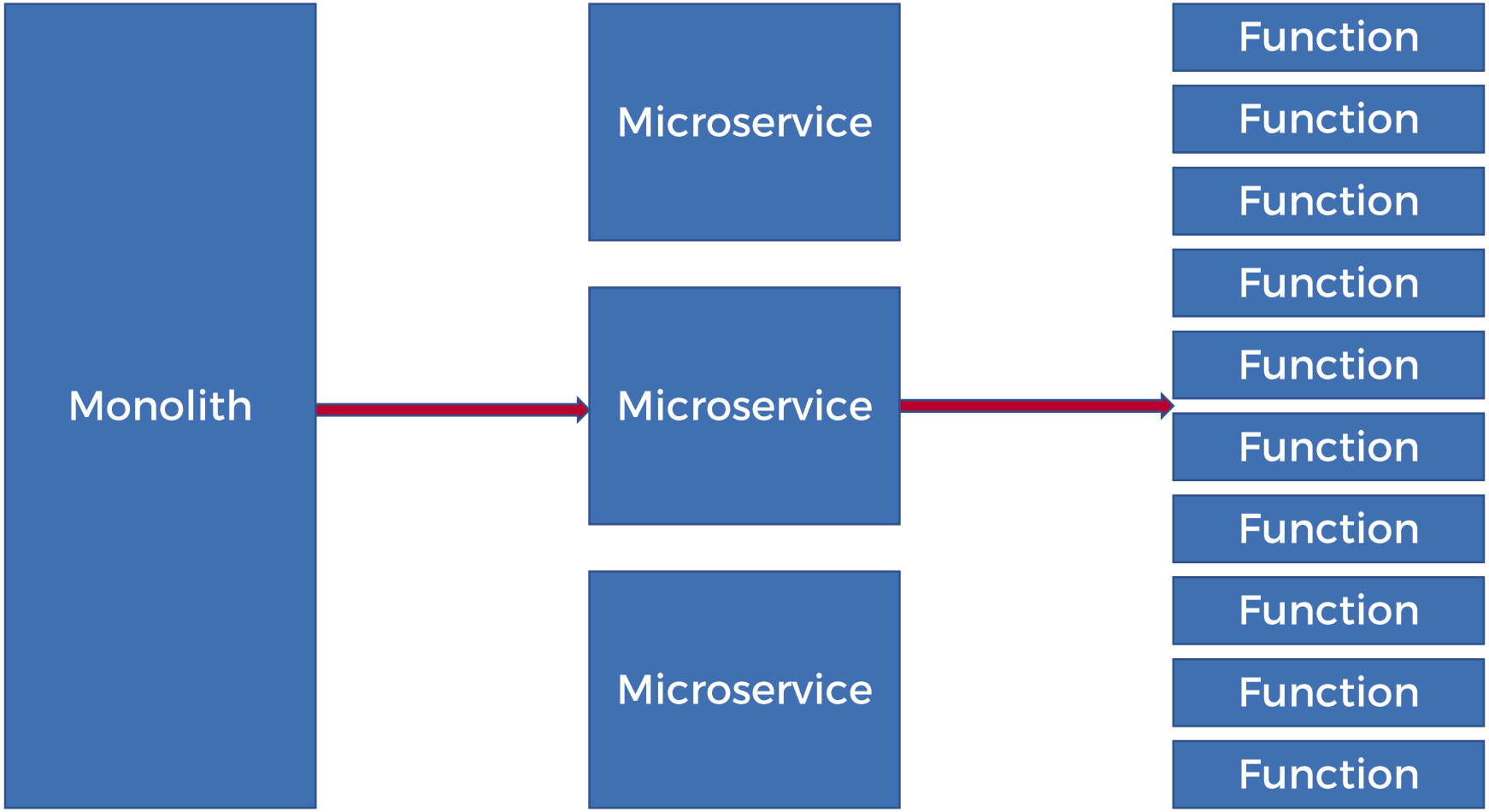
	DIY	Bare Metal	PaaS	K8s-aaS	Serverless
Hardware					
OS					
Hypervisor					
Virtual OS					
Container Orchestration					
Application Code					

Your problem

¯_(ツ)_/¯

Somebody else's problem

Credits: Matthew Gilliard


Monolith
Microservice
Microservice
Microservice
Function
Function
Function
Function
Function
Function
Function
Function
Function
Function

Google Cloud

Azure

CLOUD DEVELOPER DAYS

- ✓C#
- ✓F#
- ✓JavaScript
- ✓*some experimental*

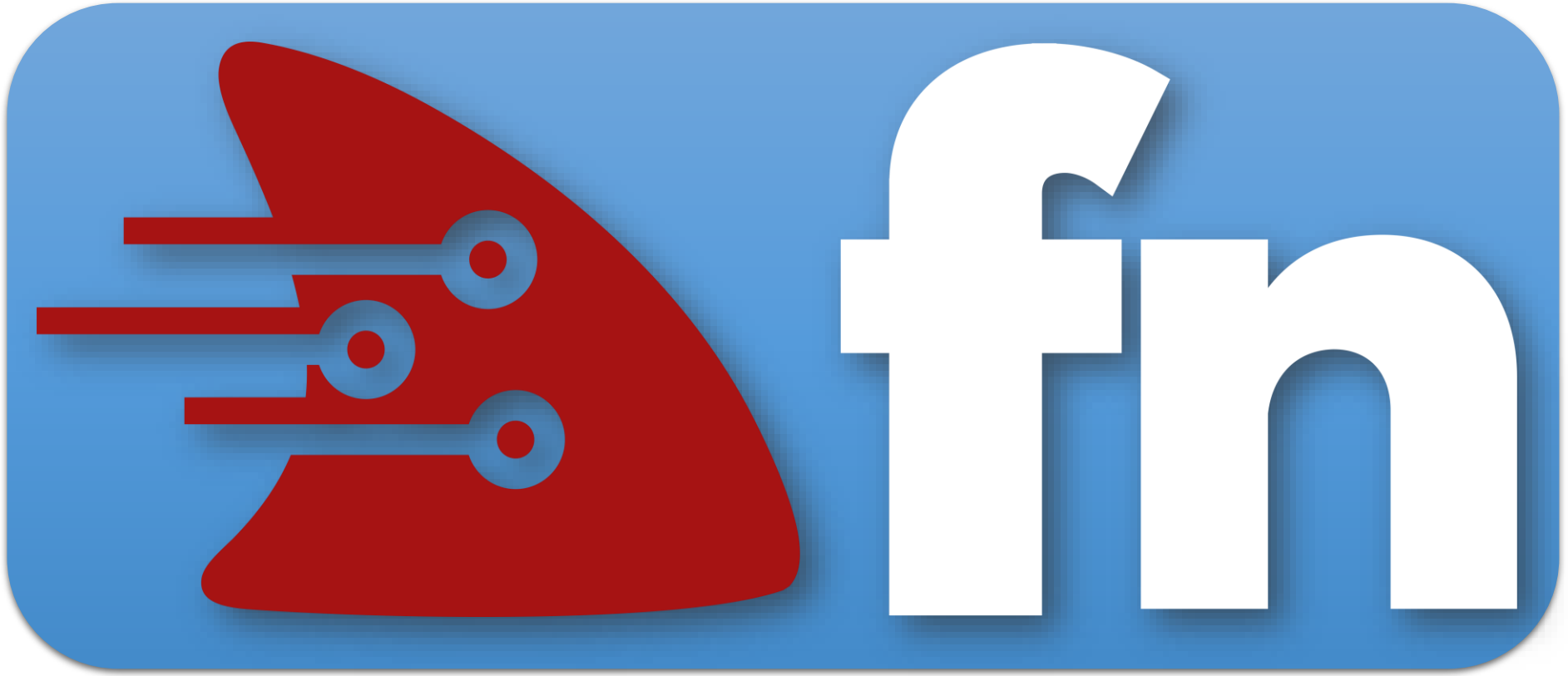
- ✓JavaScript
- ✓Python
- ✓Java
- ✓C#
- ✓Go
- ✓*some wacky workarounds*

Google Cloud

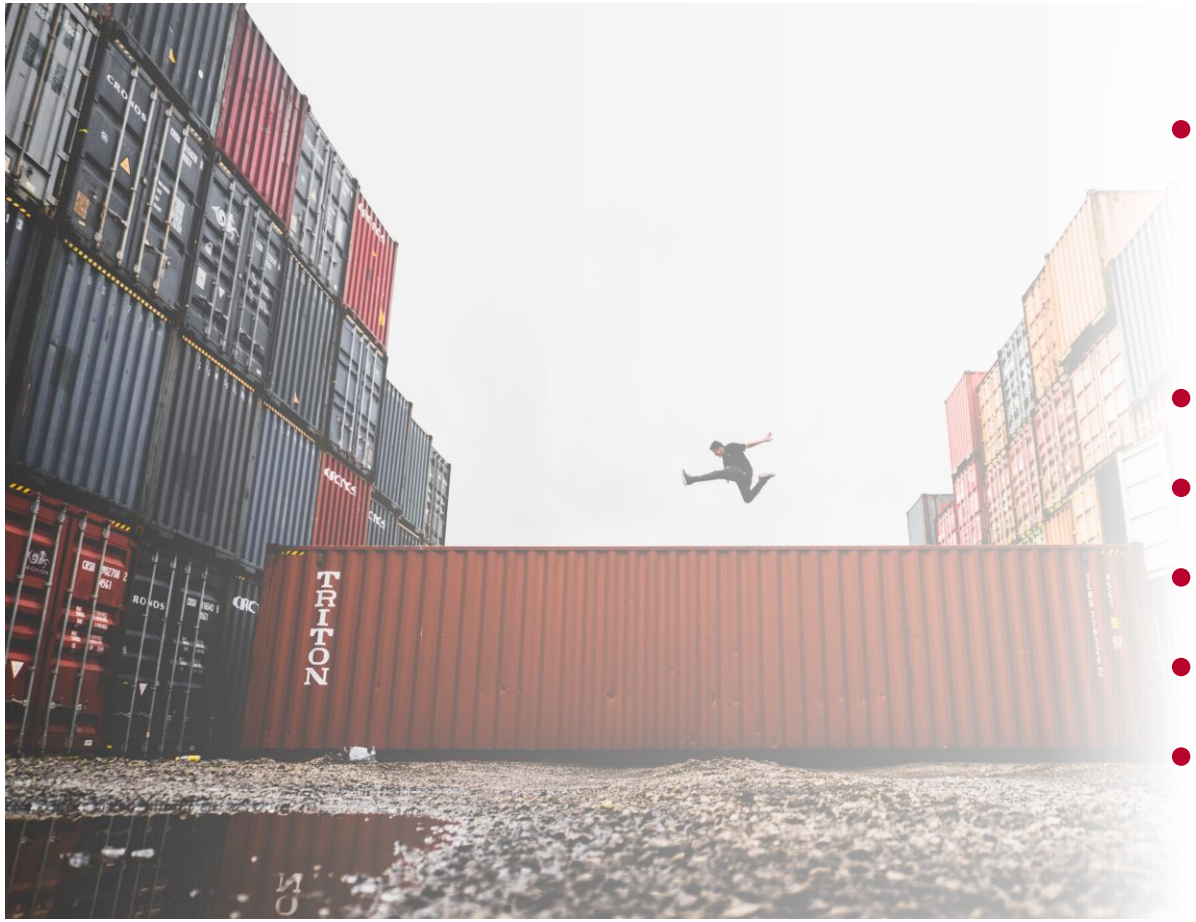
CLOUD DEVELOPER DAYS

✓JavaScript
✓*nothing else*

That’s it?!?


fn

What is Fn Project?



- Independent **open-source** serverless compute platform

- Not tied to any cloud vendor
- Can be run on premises
- Supported by Oracle
- Containers are primitives
- Strong enterprise in mind

What is a function container?

- Sandboxed process
- Short running
- Event-driven
- Stateless (-ish)

Anatomy of an Fn function

- Small chunk of code wrapped in a container
- Gets input from http-stream and environment
- Sends output to http-stream
- Logs to STDERR / syslog

FDK (Function Development Kit) support

Go

Java

Node.js

Python

Ruby

C#

The Fn Server

- Runs in a container also
- Handles as an API gateway
- Exposes REST interface
- Storing logs

The Fn CLI

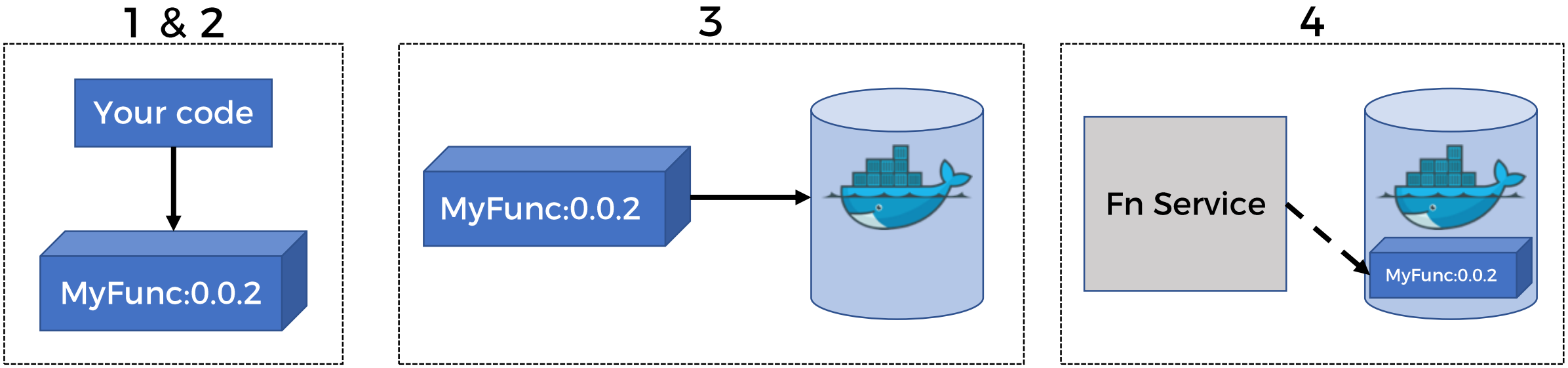
- Of course we have a CLI!
- Used to interact with Fn Server
- Initialization
- Deployment
- Invokation

Initialization of functions

- Calling **fn init** to create a boilerplate in a folder
- Does not do anything on the server
- Boilerplate includes: **Dockerfile**, **func.yaml** & **code**

How deployment works

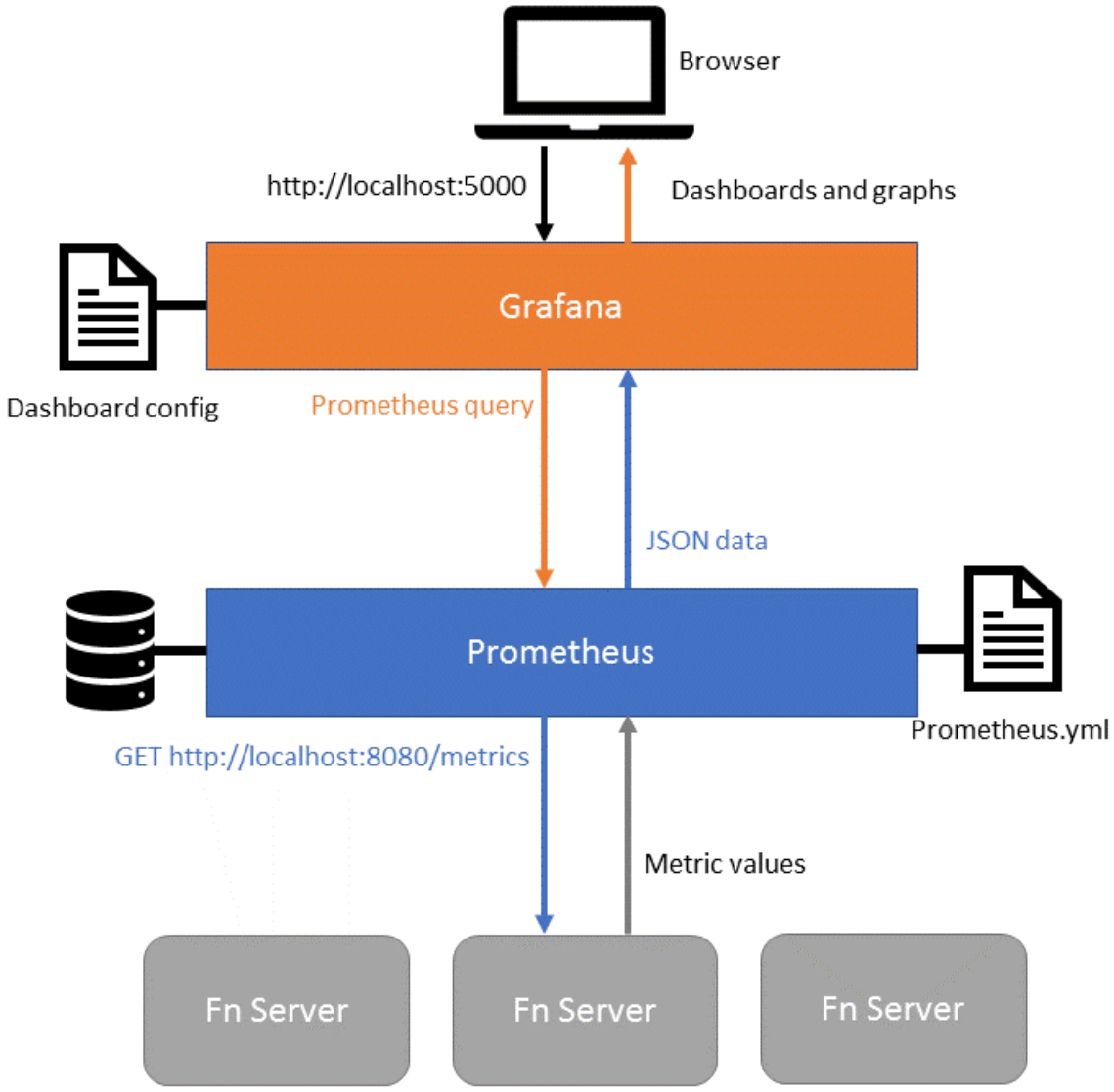
1. Bumps function version
2. Builds container
3. Pushes it to registry
4. Creates or update function & trigger in server



Invocation of functions

- From the CLI
- With HTTP request
- From the UI

DEMO



CLOUD DEVELOPER DAYS

See metrics with Grafana & Prometheus

Fn usage
Zoom Out
Last 5 minutes
Queued: 297
Running: 5
Completed: 266
Failed: 7
Total queued
Total queued 297
Total running
Total running 5.00
Total completed
Total completed 266
Total queued
Total failed 7
Queued
/hello1 134
/hello2 99
/hello3 64
Running
/hello3 1
/hello1 2
/hello2 2
Completed
/hello1 148
/hello2 87
/hello3 31
Failed
/hello2 4.00
/hello1 3.00

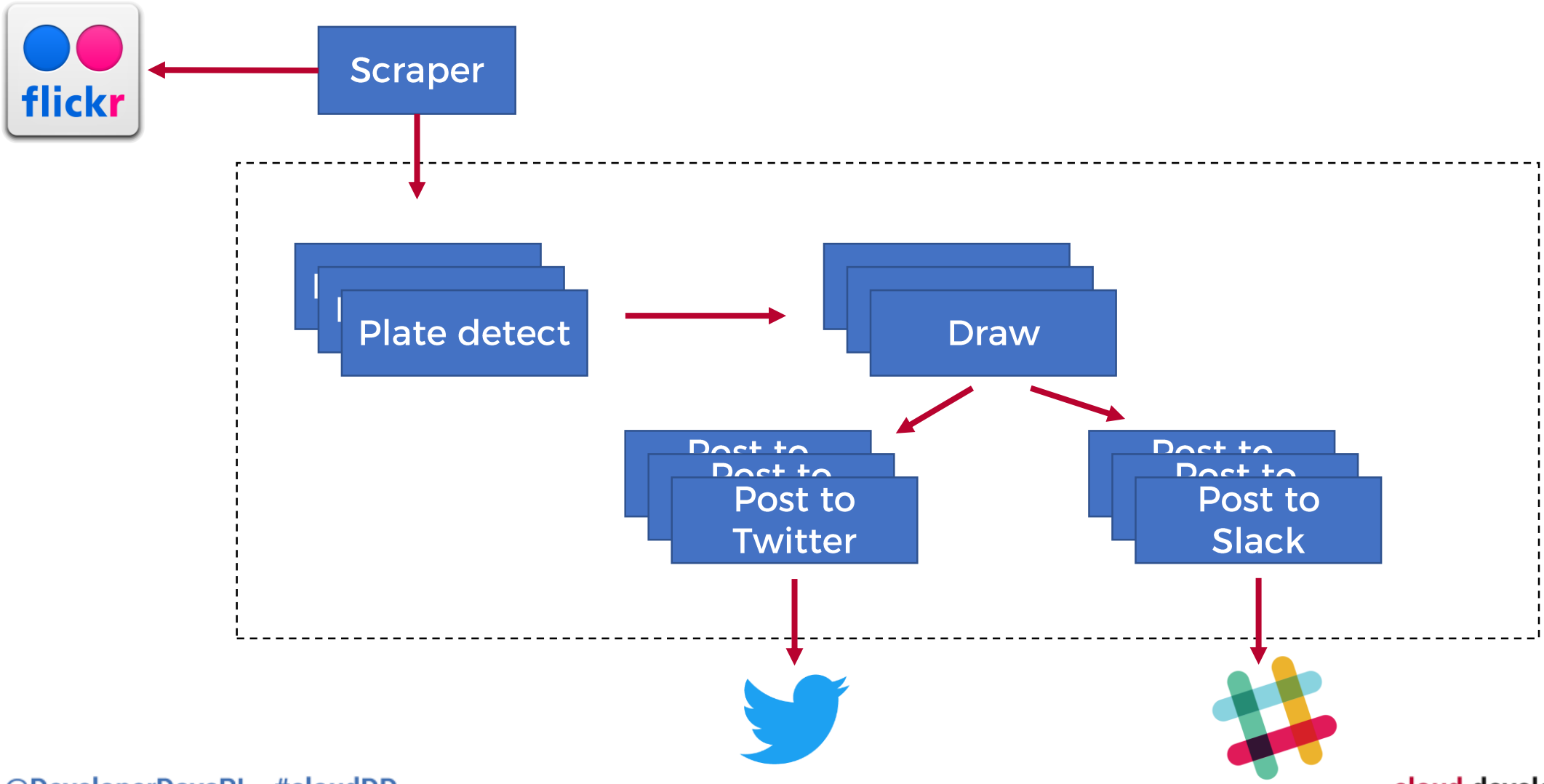
Using it in Kubernetes

- Recommended way of deployment for production
- Helm chart available on GitHub

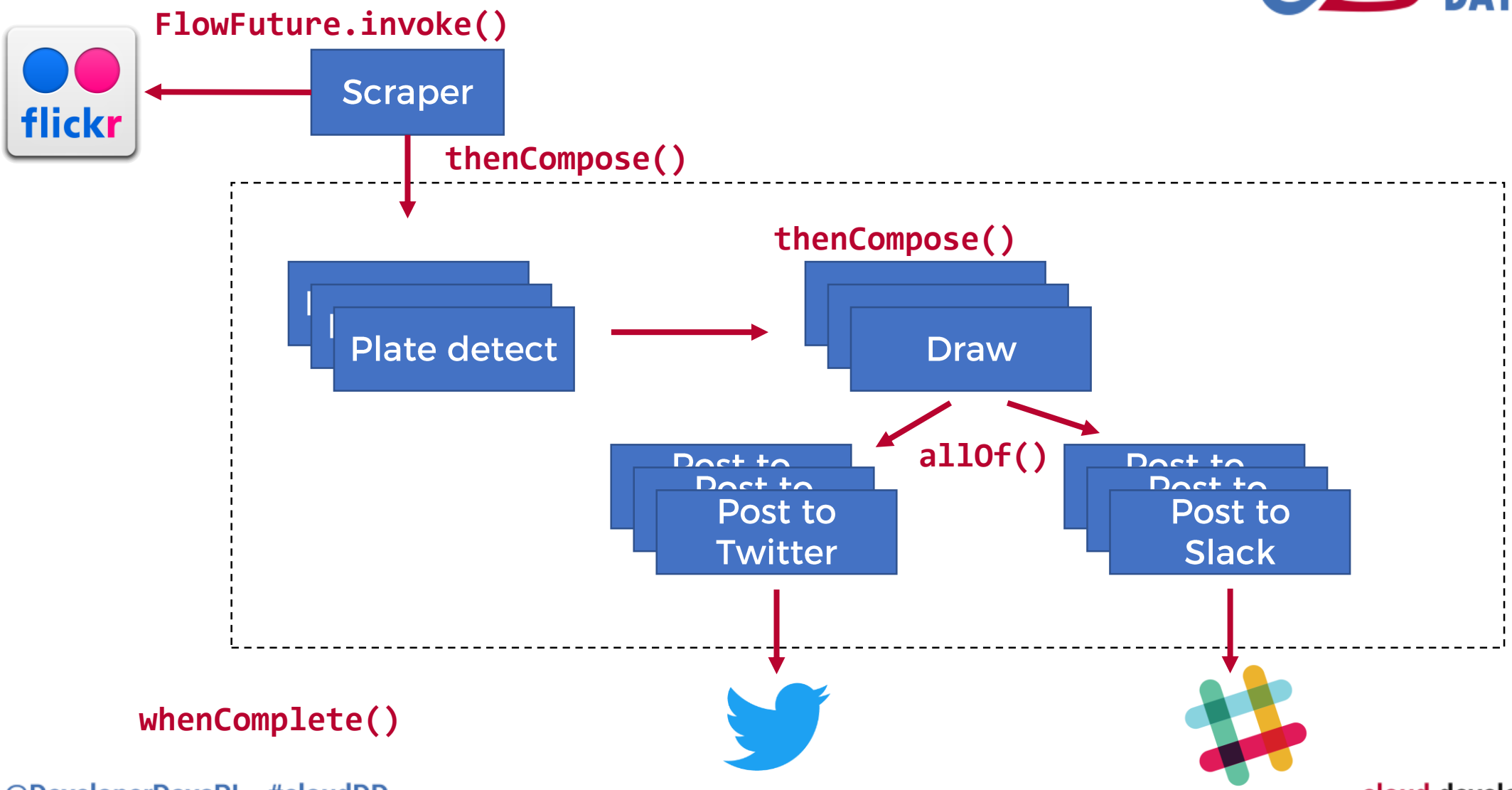
Fn Flow

- Building scalable distributed applications out of functions
- Flows are functions also
- Support complex parallel processes with error handling, which is testable
- Flow functions scale as normal functions
- Currently supports Java, hopefully more to come

Fn Flow | Licence plate example



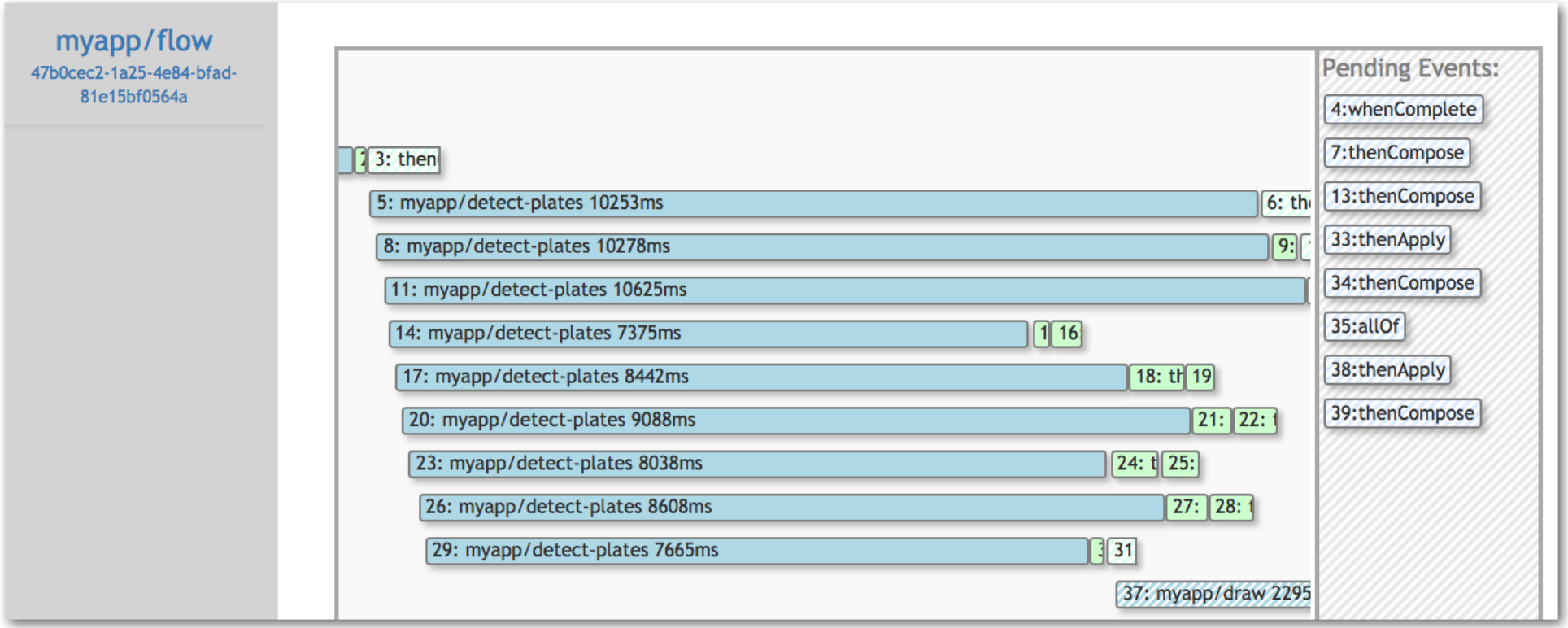
Fn Flow | Licence plate example



```
1   public class VistaFlow {
2     public void handleRequest(ScrapeReq input) throws Exception {
3
4     FlowFuture<ScrapeResp> scrapes = currentFlow().invokeFunction("scraper");
5
6     scrapes.thenCompose(resp -> {
7       List<FlowFuture<?>> pendingTasks = results.stream().map(scrapeResult -> {
8
9         String id = scrapeResult.id;
10
11        return currentFlow().invokeFunction("detect-plates").thenCompose((plateResp) -> {
12
13          return currentFlow().invokeFunction("draw").thenCompose((drawResp) -> {
14
15            return currentFlow().allOf(
16              currentFlow().invokeFunction("twitter"),
17              currentFlow().invokeFunction("slack"));
18          });
19        });
20      });
21
22    }).whenComplete((v, throwable) -> {
23        if (throwable != null) {
24          postMessageToSlack("An Error Occurred.");
25        } else {
26          postMessageToSlack("Finished Scraping.");
27        }
28      });
29    }
30  }
```

Fn Flow | Licence plate example

Fn Flow UI


myapp/flow
47b0cec2-1a25-4e84-bfad-81e15bf0564a
3: then
5: myapp/detect-plates 10253ms
6: th
8: myapp/detect-plates 10278ms
9:
11: myapp/detect-plates 10625ms
14: myapp/detect-plates 7375ms
1
16
17: myapp/detect-plates 8442ms
18: th
19
20: myapp/detect-plates 9088ms
21:
22:
23: myapp/detect-plates 8038ms
24: t
25:
26: myapp/detect-plates 8608ms
27:
28:
29: myapp/detect-plates 7665ms
31
37: myapp/draw 2295
Pending Events:
4:whenComplete
7:thenCompose
13:thenCompose
33:thenApply
34:thenCompose
35:allOf
38:thenApply
39:thenCompose

DEMO

Please rate this session using

Event Master
Mobile App

login.developerdays.pl

at the booth
in Lobby

Where to find me:

Todor Todorov

e: todor@todorov.bg

b: www.todorov.bg

t: www.twitter.com/totollygeek

l: www.linkedin.com/in/totollygeek

g: www.github.com/totollygeek

STRATEGIC SPONSORS

IBM

SPYROSOFT

GOLD SPONSORS

Demant Technology Centre

aruba CLOUD

ABB

Fast Reports
Reporting must be Fast!