

Break New Ground

Serverless Architectures and Patterns in Action

Abhishek Gupta
Senior Product Manager
Oracle Cloud Infrastructure—Serverless

@abhi_tweeter

March 15, 2019



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Agenda

- Hello Serverless
- Open source *fnproject* and *Oracle Functions*
- “... *in action*” – use cases + demos
- QnA

For slides and code...

<https://github.com/abhirockzz/oraclecode2019>



Serverless



**ORACLE
CODE**



Rod Johnson
@springrod

Follow

How can serverless have server errors? Just tried to invoke a Lambda through HTTP and got {"message": "Internal server error"}

10:16 AM - 2 Mar 2019

182 Retweets 755 Likes



44

182

755



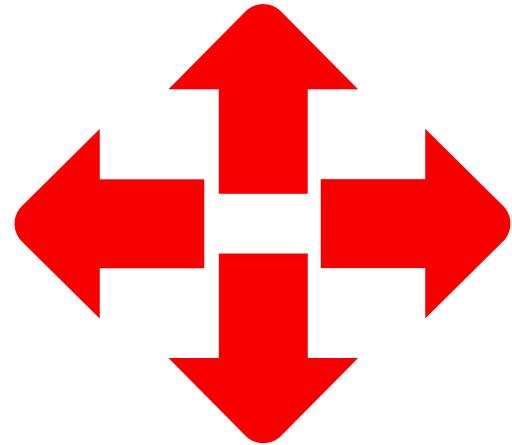
Serverless characteristics



There are servers, but
don't worry about them
(provisioning, patching
etc.)



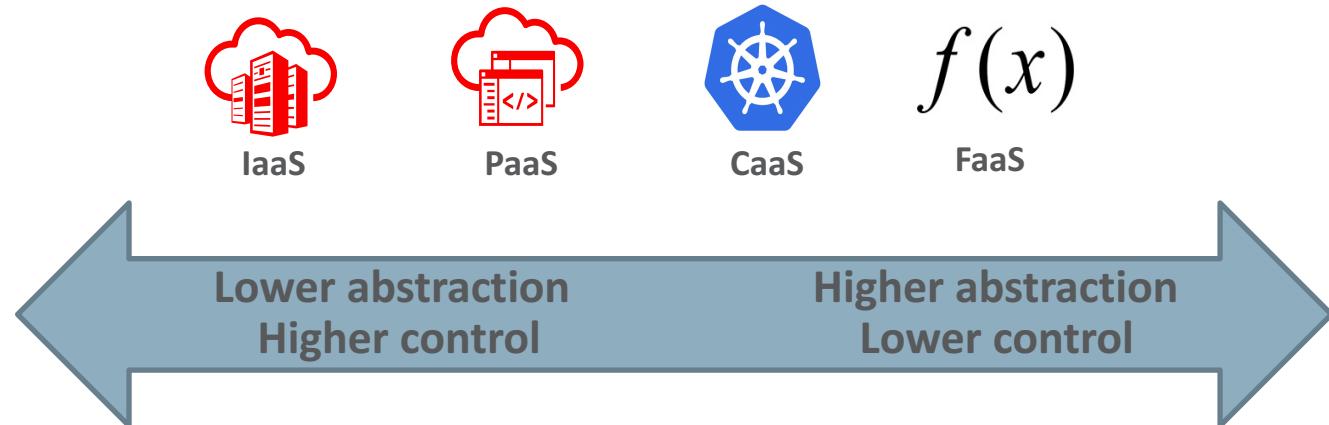
Cost effective – pay only
for use



Scale (in and out)
automatically

Functions

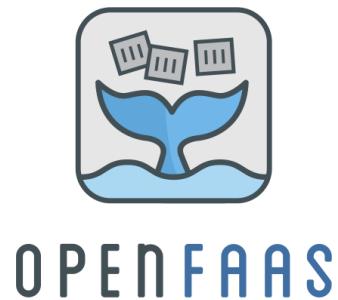
The serverless application programming model



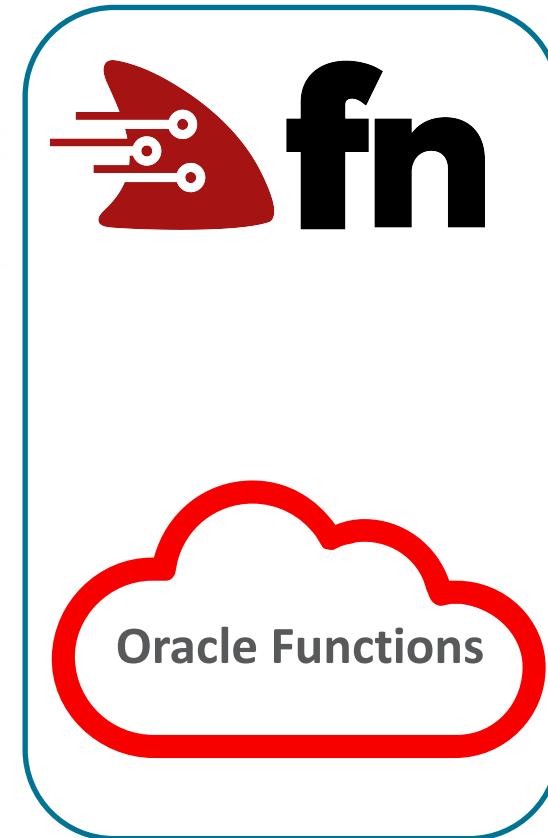
- Simple piece of code that does one job
- Event driven
- More often than not
 - Stateless
 - Short
- All the *Serverless* goodness!
 - Don't worry about underlying infra
 - Pay only when code executes
 - Auto scale

FaaS-ing options

Open source



Cloud offerings



Google Cloud Functions

and lots more....

Fn Project & Oracle Functions



Fn—An open source Functions Platform



- **Open Source**—no vendor lock-in
- **Docker Based**—leverage Docker ecosystem
- **Platform Independent**—laptop, server, cloud
- **Scheduler Independent**—deploy to Kubernetes, Swarm, Mesos, etc.
- **Approachable**—easy for new users, low level controls for advanced users

Introducing Oracle Functions

Coming soon (currently in Limited Availability)

Oracle Functions

Functions-as-a-Service

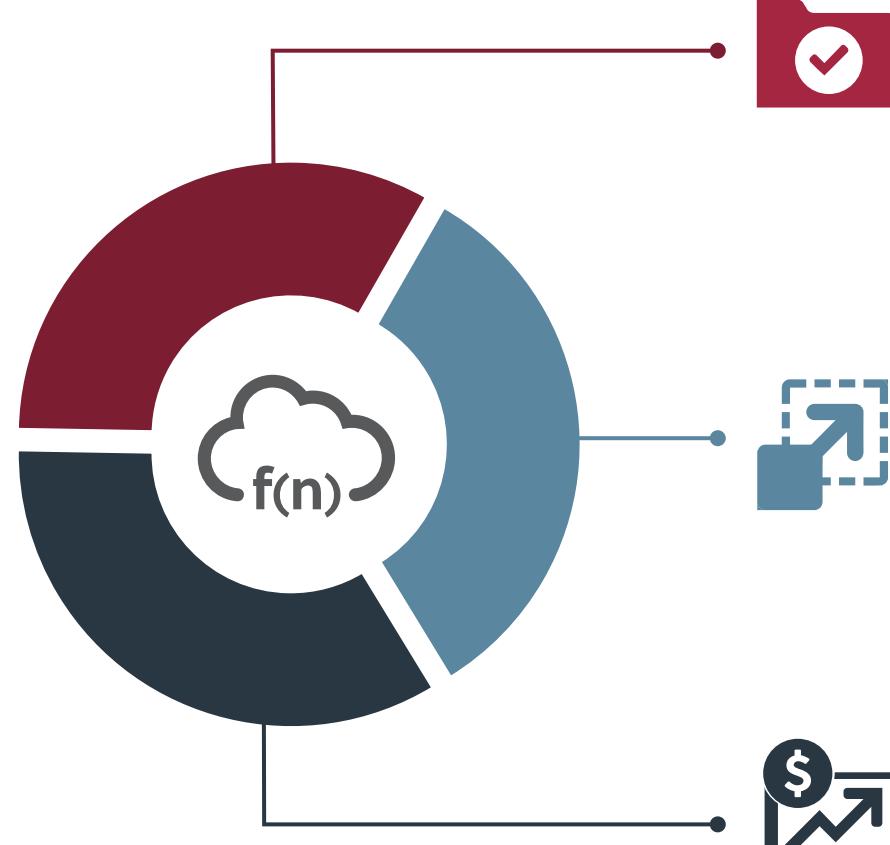
Oracle Cloud Integrated

Container Native

Open Source Engine

Multi-tenant

Secure



Pay Per Use

Pay for execution, not for idle time

Autonomous

Platform auto-scales functions
No servers to provision,
manage

No Lock-in

Built on open-source Fn Project and Docker

Key concepts



**ORACLE
CODE**

A Function

```
package com.example.fn;

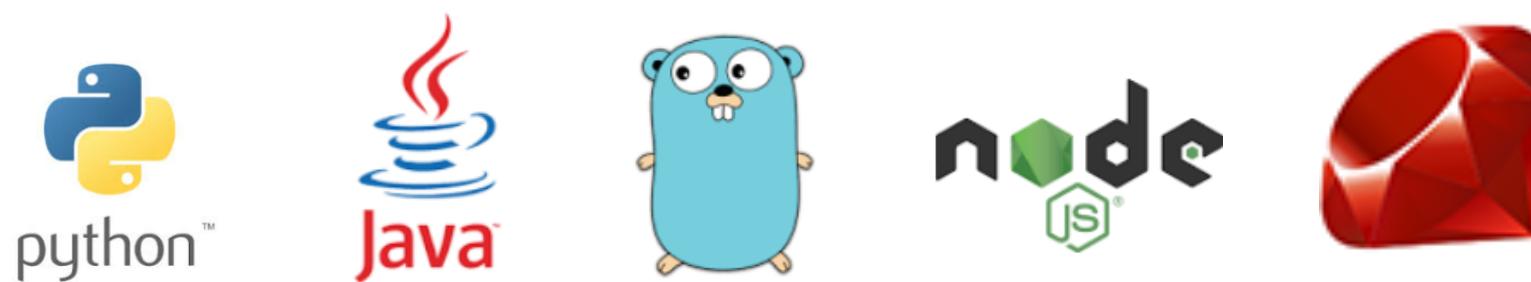
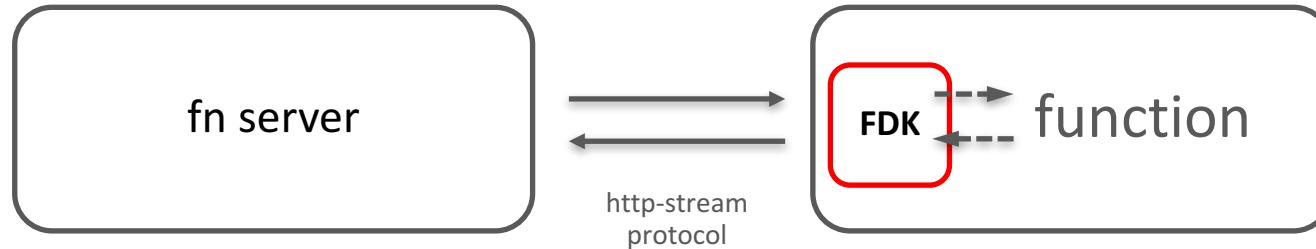
public class HelloFunction {

    public String handleRequest(String input) {
        String name = (input == null || input.isEmpty()) ? "world" : input;

        return "Hello, " + name;
    }

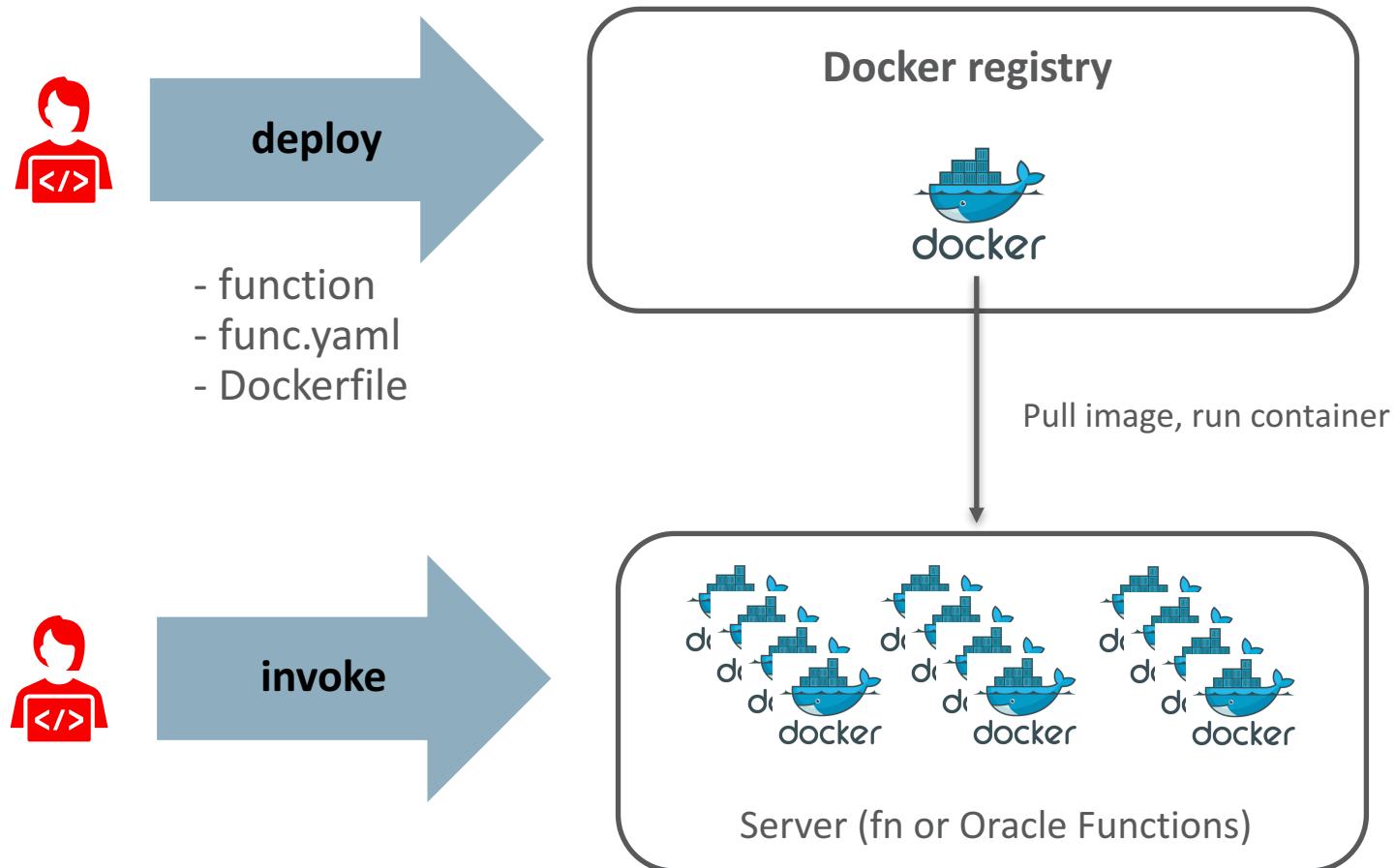
}
```

Function Development Kits (FDKs)



[FDK documentation](#)

deploy and run



CLI

| ENVIRONMENT VARIABLES | |
|-----------------------|---|
| FN_API_URL | <i>Fn server address</i> |
| FN_REGISTRY | Docker registry to push images to, use username only to push to Docker Hub - [[registry.hub.docker.com/]USERNAME] |
| GENERAL COMMANDS | |
| version | Display Fn CLI and Fn Server versions |
| help, h | Shows a list of commands or help for one command |
| DEVELOPMENT COMMANDS | |
| build, bu | Build function version |
| bump, bm | Bump function version |
| deploy, dp | Deploys a function to the functions server (bumps, build, pushes and updates functions and/or triggers). |
| init, in | Create a local func.yaml file |
| invoke, iv | Invoke a remote function |
| migrate, m | Migrate a local func.yaml file to the latest version |
| push, p | Push function to docker registry |
| MANAGEMENT COMMANDS | |
| config, cf | Set configuration for an object |
| create, c | Create a new object |
| delete, d | Delete an object |
| get, g | Get an object to retrieve its information |
| inspect, i | Retrieve properties of an object |
| list, ls | Return a list of created objects |
| unset, un | Unset elements of a created object |
| update, up | Update a created object |
| use, u | Select context for further commands |
| SERVER COMMANDS | |
| build-server | Build custom Fn server |
| start | Start a local Fn server |
| stop | Stop a function server |
| GLOBAL OPTIONS | |
| --context value | Use --context to select context configuration file [\$FN_CONTEXT] |
| --registry value | Use --registry to select registry |
| --verbose, -v | Use --verbose to enable verbose mode for debugging |



“HelloWorld” demo



**ORACLE
CODE**

fn - hello world demo

- 1. fn start**
- 2. fn init --runtime go --name hellofunc**
- 3. fn use context default**
- 4. fn create app oracodemo**
- 5. fn -v deploy --app oracodemo --local**
- 6. fn invoke oracodemo hellofunc**

Moving to the cloud - Oracle Functions

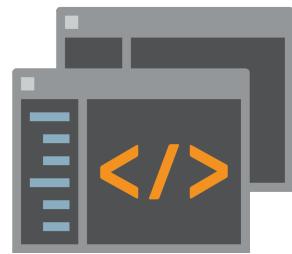
1. **fn start**
2. **fn init --runtime go --name hellofunc**
3. **fn use context <name>**
4. **fn create app <*additional info*> oracodemo**
5. **fn -v deploy --app oracodemo --local**
6. **fn invoke oracodemo hellofunc**

Serverless Use Cases & Patterns



**ORACLE
CODE**

Backend systems



- APIs
- Web apps
- Simple apps

Event-Driven systems



- Event driven apps
- Stream Processing

Integration



- Webhooks
- Workflows

Automation

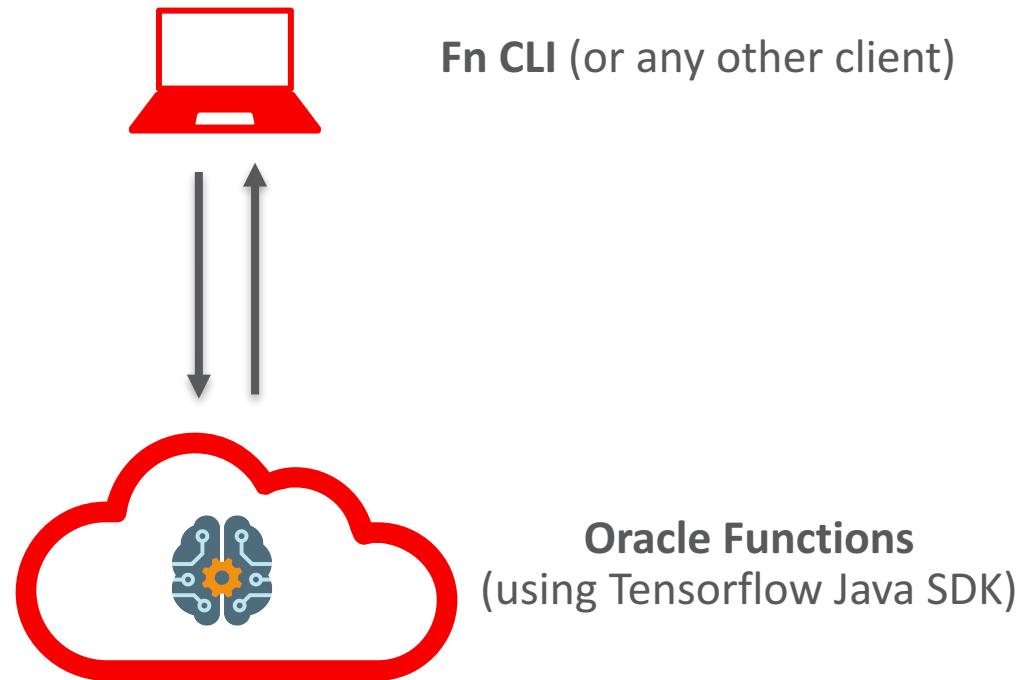


- IaaS monitoring
- Scheduled/Batch jobs

Backend systems

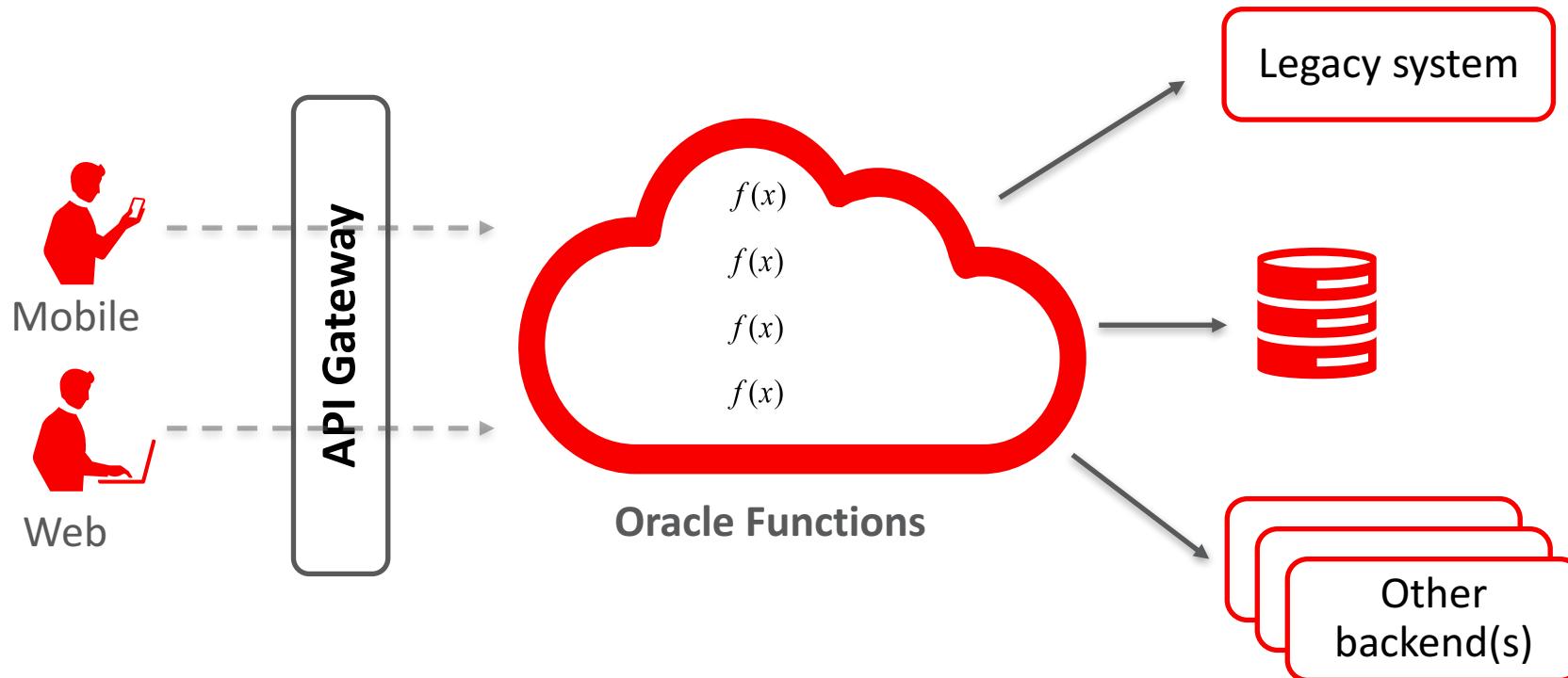
**ORACLE
CODE**

Single purpose apps

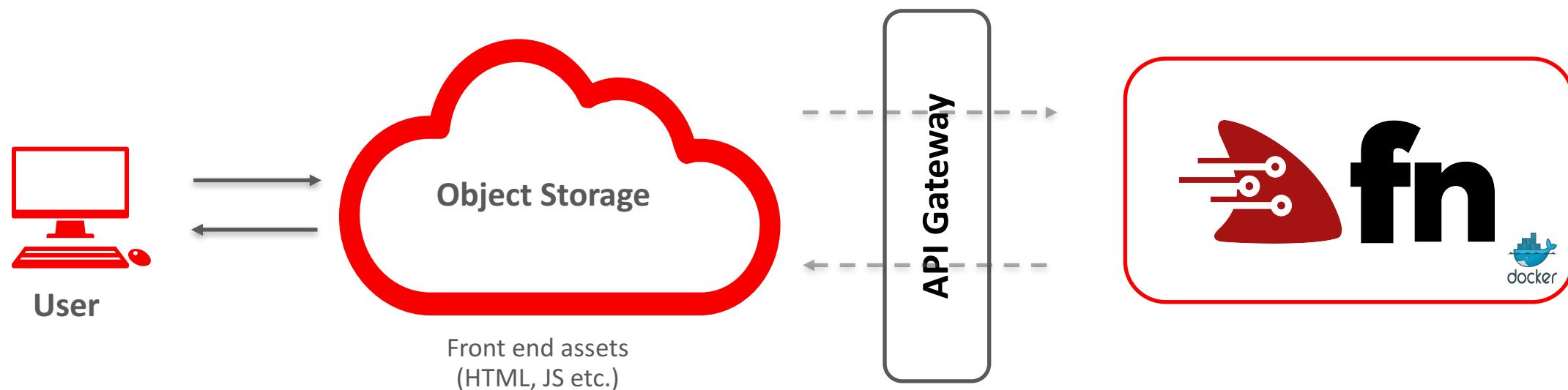


Tensorflow Image Classification

APIs (data services)



FaaS APIs + ‘friends’ = Serverless Web apps

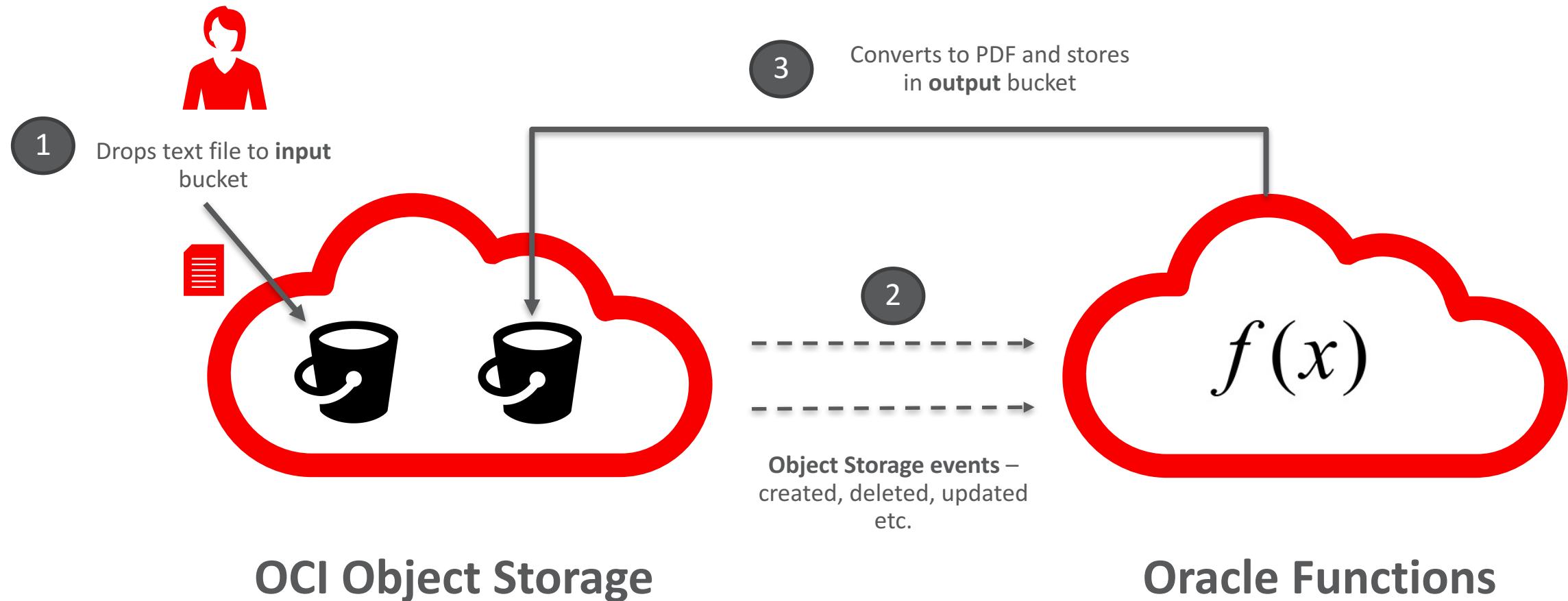


Event-Driven systems



ORACLE
CODE

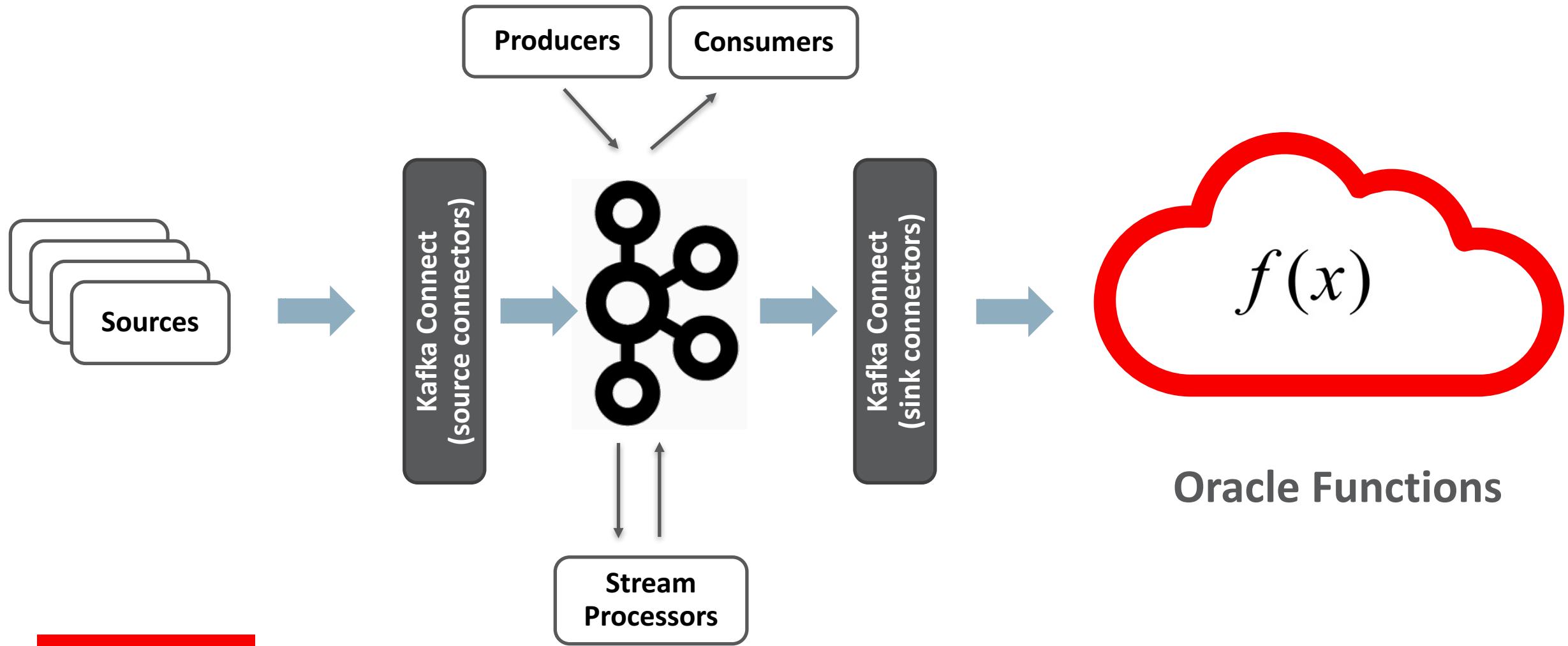
Asynchronous processing



Stream Processing

- Built to handle continuous (streams of) data
 - Durable/persistent
 - Possible to replay and reprocess data (aka time travel)
 - Message Ordering, low latency, different guarantees
- Amazon Kinesis, Apache Kafka etc.
- Streaming-FaaS integration
 - Native – tight coupling b/w streaming and FaaS platforms
 - vendor specific e.g. Kinesis -> Lambda
 - Custom – roll your own
 - vendor agnostic, flexible e.g. Kafka -> FaaS
 - Need a *server* component

Kafka -> Kafka Connect -> FaaS





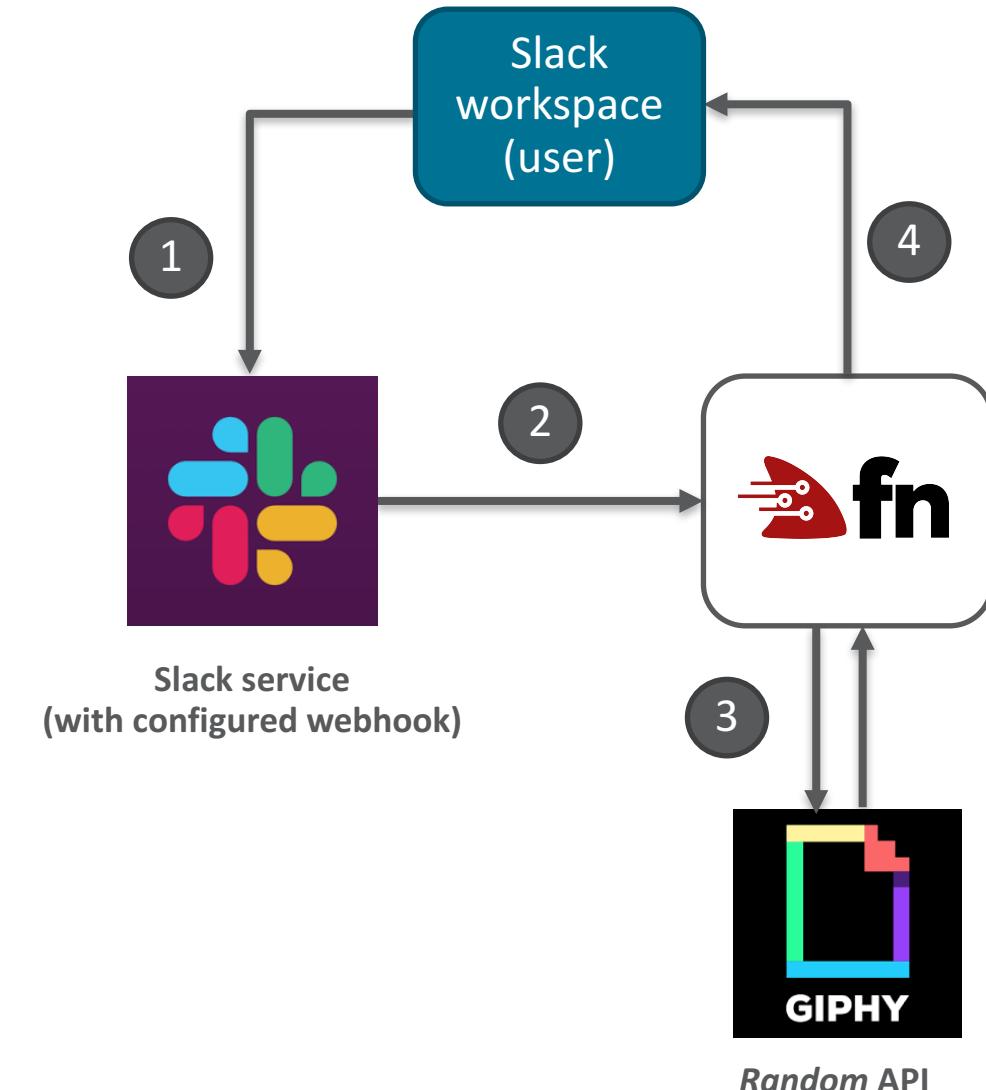
Integration



**ORACLE
CODE**

Webhooks

- [Wikipedia](#)
 - “user-defined HTTP callbacks”
 - “usually triggered by some event”
- Event driven on *steroids* – trigger functions using (almost) *any* event source
- Easily integrate with third party systems like GitHub, Slack, Twilio etc.



Serverless Jenkins Pipelines with Fn Project

The [Jenkinsfile-Runner-Fn](#) project is a [Fn Project](#) (a container native, cloud agnostic serverless platform) function to run Jenkins pipelines. It will process a GitHub webhook, git clone the repository and execute the Jenkinsfile in that git repository. It allows scalability and pay per use with zero cost if not used.

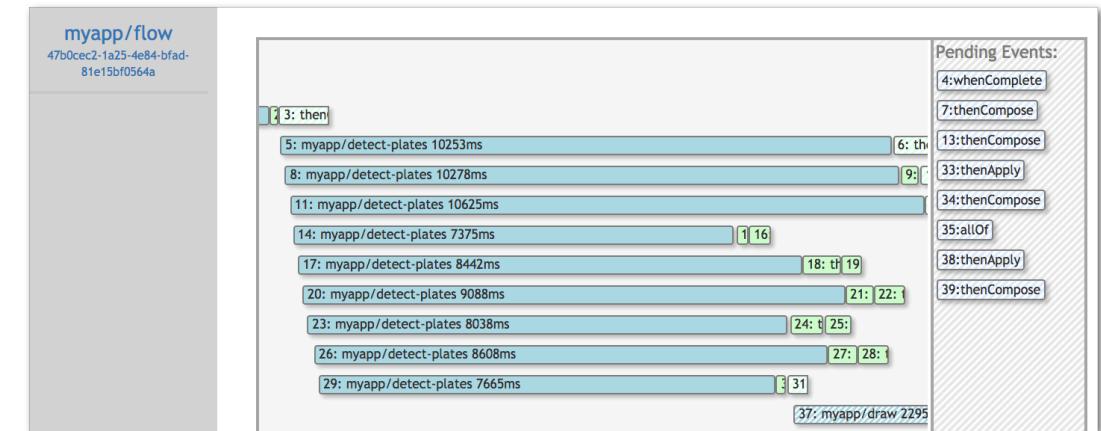
This function allows **Jenkinsfile** execution without needing a persistent Jenkins master running in the same way as [Jenkins X Serverless](#), but using the Fn Project platform (and supported providers like [Oracle Functions](#)) instead of Kubernetes.



<https://blog.csanchez.org/2019/02/19/serverless-jenkins-pipelines-with-project-fn/>

Function Orchestration

- Problem(s)
 - How do I integrate functions to build complex apps ?
 - Orchestrate their behavior
 - Manage intermediate state
- Options - AWS Step Functions, Azure Durable Functions, Fn Flow etc.
- Fn Flow
 - Build long-running functions with rich sets of language-specific primitives including fork-join, chaining, delays and error handling
 - Supports complex parallel processes that are readable and testable (including unit tests) with standard programming tools



[Fn Flow Java User Guide](#)

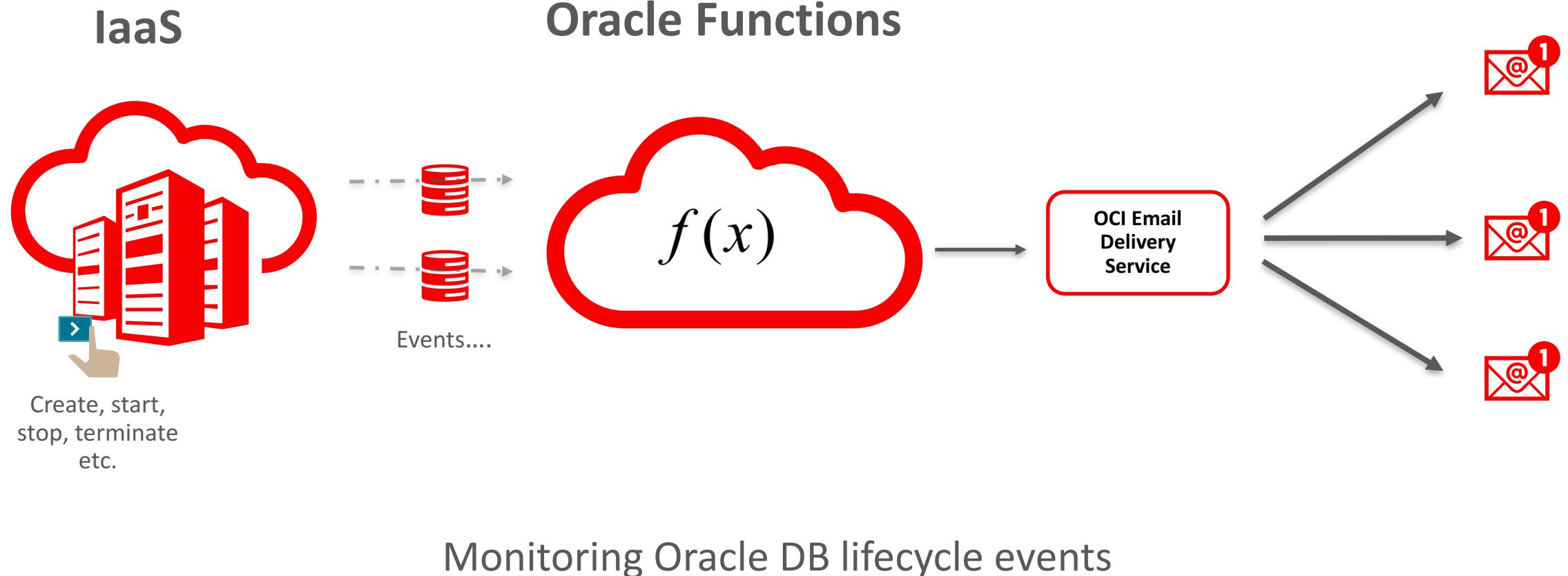


Automation



**ORACLE
CODE**

Infra monitoring



Scheduled Jobs

- Configure and forget!
- Good for automating *repetitive* tasks
 - Reminders, alerts (email, SMS etc.)
 - Backups, import/export
 - etc.



Tips

- *SRP* - try to do **one** thing well
- Factor in cold start times
 - may or may not be critical, depending on the use case
- Externalize state
 - Do not depend on ‘hot’ functions
- Be careful when using databases
 - Rely on your Serverless friends
- Avoid chaining function calls
 - Leverage higher order orchestration services if possible

Credits

- Our engineering teams
- Customers and community
- Fellow Product Managers...



Links

- **Blog** - <https://medium.com/fnproject>
- **Twitter** - <https://twitter.com/fnproject>
- **GitHub**
 - Fn – <https://github.com/fnproject/fn>
 - FDKs - <https://github.com/fnproject/?q=fdk>
 - Fn Flow - <https://github.com/fnproject/flow>
- **Slack** - <https://fnproject.slack.com/>
- **Oracle Functions** - <https://blogs.oracle.com/cloud-infrastructure/announcing-oracle-functions>
- **GitHub repo for this talk** - <https://github.com/abhirockzz/oraclecode2019>

Thank you !

ORACLE®