

CLOUD NATIVE
FINERACT WITH
SCALABILITY

Presented by Zoltan Mezei

FinTech Track

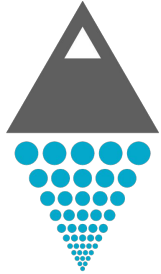


Halifax, NS

October 6-10, 2023

The Boring Slide

Zoltan Mezei - zoltan.mezei@baasflow.com



- <https://baasflow.com/>

BaaSFlow



PIVOT
DIGITAL



Apache Fineract

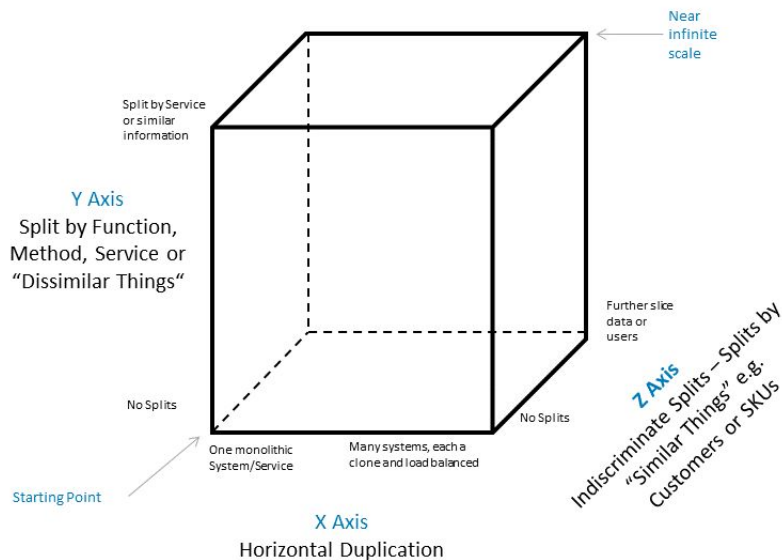
Agenda



- Previously on ApacheCon2022...
- The AKF Scale Cube
- Fineract Instance Modes
- New Helm Chart
- Managed Platform Concept
- Terraform Guardrails
- Stay tuned for ApacheCon2024...

The AKF Scale Cube and Fineract

AKF Scale Cube



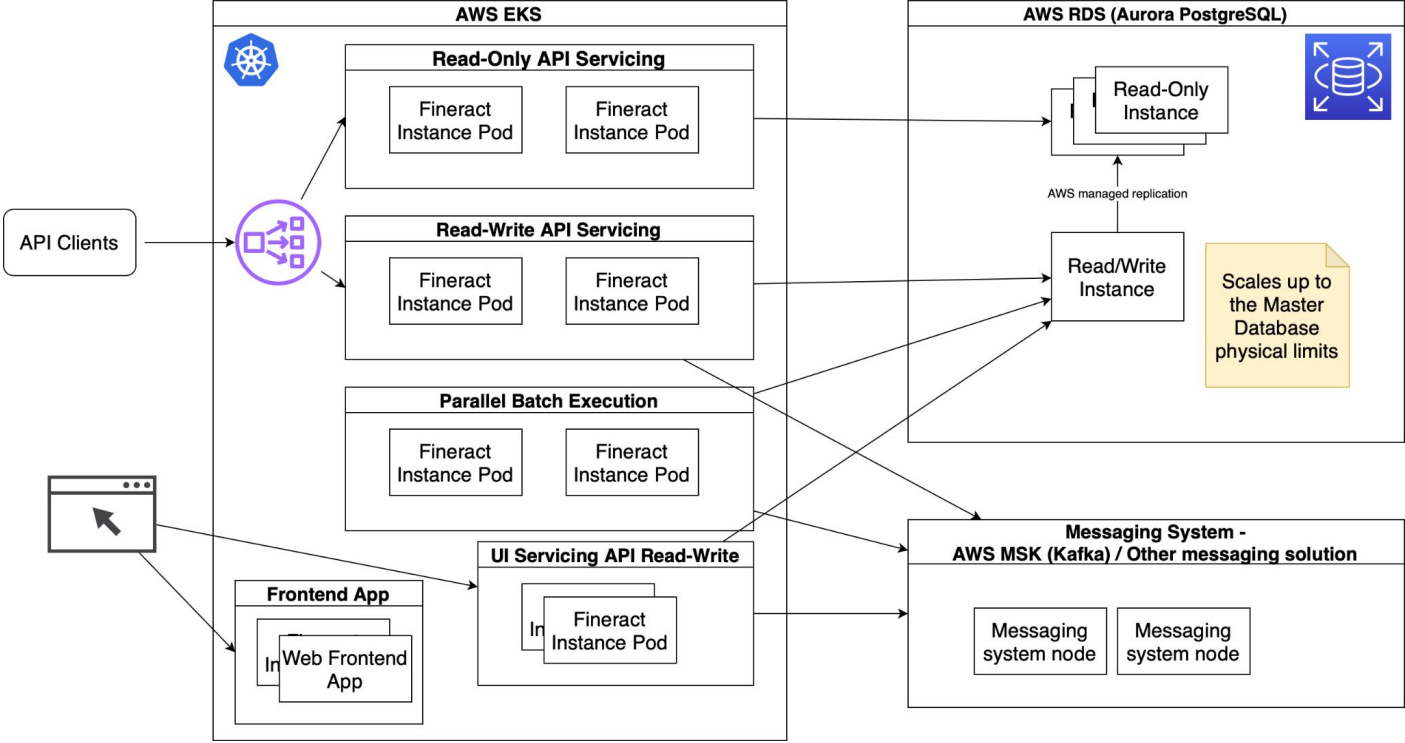
Fineract 2022

- X Axis → database replication
- Z Axis → tenants

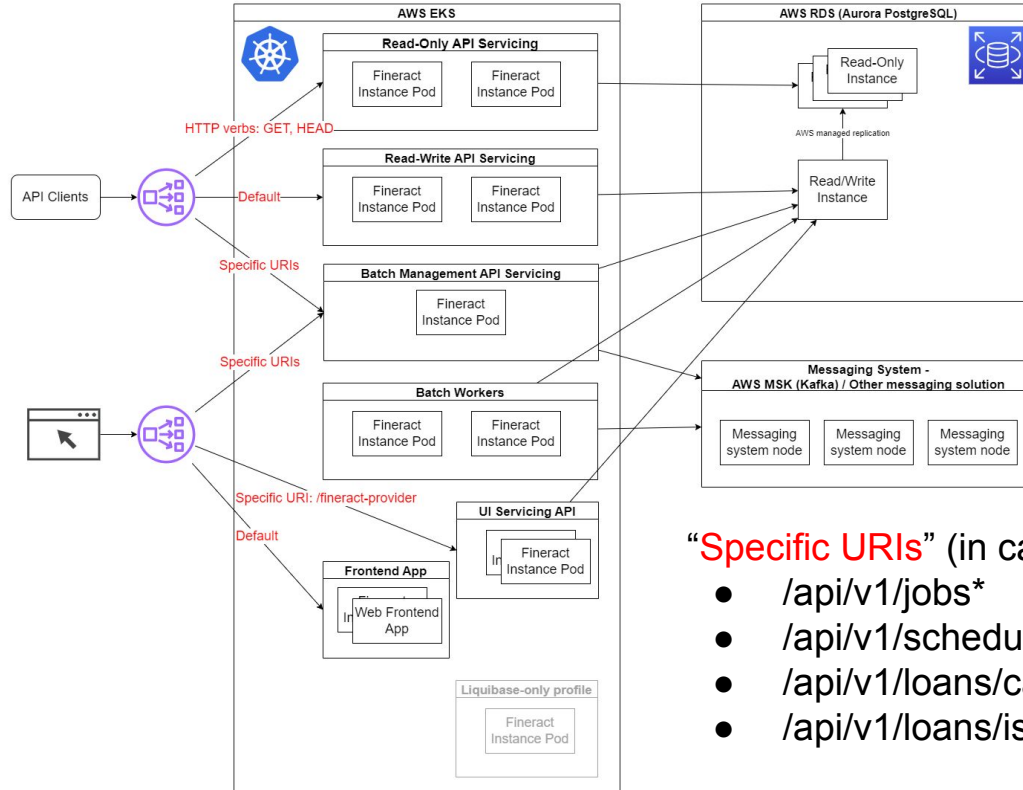
Fineract 2023

- X Axis → database replication + Deployment/StatefulSet autoscaling + batch work distribution
- Y Axis → Fineract instance modes + Liquibase-only profile
- Z Axis → tenants

Fineract Instance Modes 2022



Fineract Instance Modes 2023



“Specific URIs” (in case you are really interested):

- /api/v1/jobs*
- /api/v1/scheduler*
- /api/v1/loans/catch-up*
- /api/v1/loans/is-catch-up-running*

Fineract on Kubernetes Highlights

- New Helm Chart packed with features
 - Support for multiple Workloads in the same chart
 - Liquibase run in the pre-installation/pre-upgrade hook
 - Multiple Deployment and StatefulSet instances with different configurations (files and environment variables)
 - Autoscaling and other production-ready performance, security and availability features
 - Unified Ingress instances for the automatic request distribution
- Disclaimer: currently supports AWS EKS + AWS load-balancer controller only

Helm Chart Screenshot

```
1 app:
2   name: *name "fineract"
3   serviceAccount:
4     enabled: true
5     name: *name
6     automountServiceAccountToken: false
7     role: "arn:aws:iam::<snip>:role/<snip>-fineract-<snip>"
8   workload:
9     image:
10      registry: "<snip>.dkr.ecr.eu-central-1.amazonaws.com"
11      repository: "baasflow/fineract"
12      tag: "__gets-filled-from-the-deployment-job__"
13      imagePullPolicy: "Always"
14   preWorkloadJobs:
15     migration: <13 keys>
16   instances:
17     write:
18       instanceName: "write"
19       enabled: true
20       templates: <3 keys>
21       configMap:
22         enabled: true
23         namePrefix: *name
24         variables: <22 keys>
25       secret: <3 keys>
26       secretFiles: <1 key>
27       workloadType: Deployment
28       workload:
29         enabled: true
30         namePrefix: *name
31         resources: <2 keys>
32         javaToolOptions: "-XX:+UseContainerSupport -XX:ActiveProcessorCount=2 -XX:MaxRAMPercentage=75 -XX:+UseG1GC -Djava.security.properties=/app/java.security.override -XX:PrintFlagsFinal"
33         ports: <1 key>
34         probes: <3 keys>
35         emptyDirVolumes: <1 key>
36         initContainers: 2
37         maxKew: 1
38         hpa: <8 keys>
39         pdb: <3 keys>
40         service: <3 keys>
41         ingress: <1 key>
42         rbac: <12 keys>
43         batchManager: <12 keys>
44         batchWorkers: <12 keys>
45   ingress:
46     enabled: true
47     namePrefix: *name
48     scheme: "internet-facing"
49     groupName: *name
50     port: 8080
51     certificateArn: null
52     fqdn: "<snip>.dev.baasflow.cloud"
53
```

Performance
Tuning

Availability
Features

495 lines

Managed Platform Concept

2022 version

- Centralized management by BaaSFlow via Jenkins
- Centralized deployment by BaaSFlow in our own AWS accounts



Source: [dreamstime.com](https://www.dreamstime.com/)

Managed Platform Concept

Obstacles

- Regulatory issues
 - Outsourcing is challenging
 - Data protection is challenging
- Organizational issues
 - Clients want to manage their own infrastructure
 - Separation of concerns
- Technical issues
 - Missing features (e.g. Liquibase-only mode, data archival solution)
 - Missing configurations guidelines for production-readiness
 - Bugs (e.g. JDBC driver parameters)



Source: 123rf.com

Managed Platform Concept



2023 version

- Guardrails provided by BaaSFlow
 - Split “management” to separate roles:
 - Owner → who pays for the component
 - Manager → who configures the component
 - Client(s) → who benefits from the component
 - Implemented cost allocation methods
 - Split non-production and production environment and data types
- Terraform and Jenkins code opened to our partners
 - Not open sourced yet - but stay tuned

Terraform Code Screenshot

```
49 variable "environment_type" {
50     type = string
51     description = "The type of the environment (dev/test/prod)."
```

52 validation {

```
53         condition = contains([ "dev", "test", "prod" ], var.environment_type)
54         error_message = "environment_type validation failed"
55     }
56 }
57 variable "data_type" {
58     type = string
59     description = "The type of the data handled in the environment (non-prod, prod)."
```

60 validation {

```
61         condition = contains([ "non-prod", "prod" ], var.data_type)
62         error_message = "data_type validation failed"
63     }
64 }
65
```

Conclusion and future plans



Conclusion

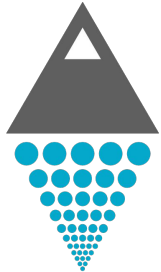
- The improvements contributed by BaaSFlow enables wider adoption of Fineract via improving the software's non-functional capabilities

Future plans

- Working on better tenant management, separation, and security
- Adding archival capabilities

Thank you

Zoltan Mezei
zoltan.mezei@baasflow.com



To learn more about
BaaSFlow and our upstream
contributions.

www.baasflow.com

