

26 June 2020

Crux and Clamp - Microservices accelerators

Building microservices with sensible choices, right foundations and speed

For information and discussion purposes only, and not to be binding document.

Only for the use of intended recipients. Do not share without prior permission.

©2020 Sahaj Software.

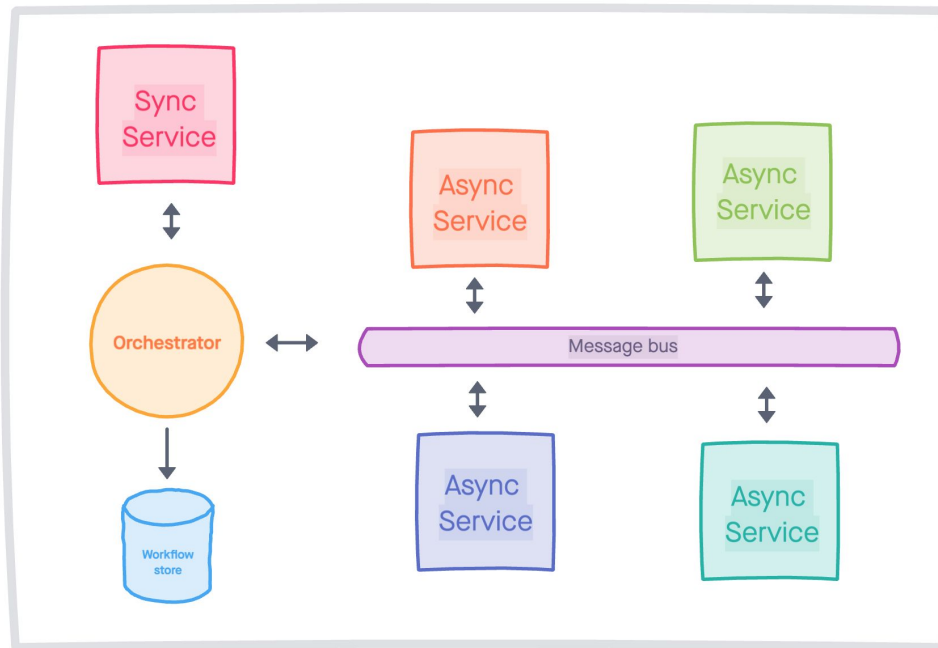
CLAMP - A microservices orchestrator

Orchestrating microservices over synchronous and asynchronous communication protocols

Approach and design

A production tested design

- A design and approach that has been used on multiple platforms in production.
- Seamless orchestration at scale with little overhead.
- Built primarily as a microservices orchestrator. Can facilitate workflow management.
- Cleaner architecture by decomposing complexities related to inter-service communication. Flows are centrally managed and tracked, which lends to readability.



Batteries included

A lightweight and opinionated approach to microservices architecture & orchestration

■ Workflow definition

A lightweight workflow management via API using JSON structure. Support for workflow creation and updates dynamically.

■ Payload transformation

Simple payload transformation between steps for JSON payloads. Past request and response content as well as metadata available as context to design service payloads.

■ Inline logic & branching

Simple branching and conditional logic support. Simplicity keep business logic out and provides required flexibility to introduce multiple execution paths.

■ Multi protocol communication

Support for synchronous and asynchronous communication between services. Message queues like RabbitMQ and HTTP transport have first class support with the orchestrator.

■ Request profiling

Basic request profiling and time distribution of a business transaction available with workflow metadata. Allows monitoring system and dashboard to plot and track time distribution between individual steps

■ Stateless & scalable

Simple, stateless and scalable structure backed by a persistent store. The orchestrator can scale and manage high throughput with ease with minimal overhead.

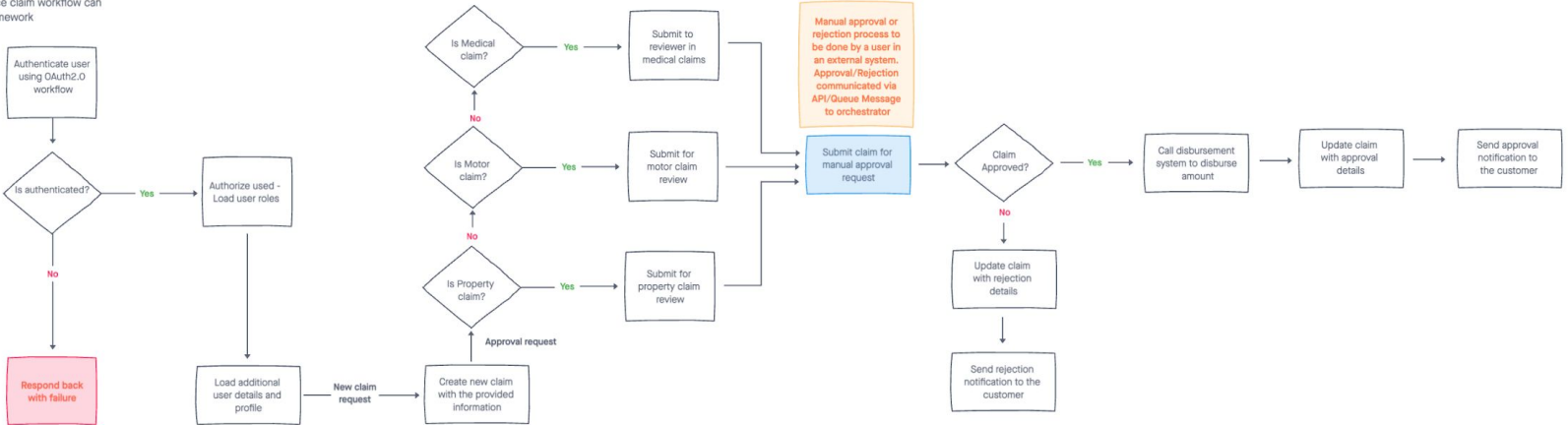
DEMO WORKFLOW

A workflow for insurance claim settlement

CLAMP - Motor insurance claim workflow

A quick demo of how insurance claim workflow can be controlled using clamp framework

START HERE ➡



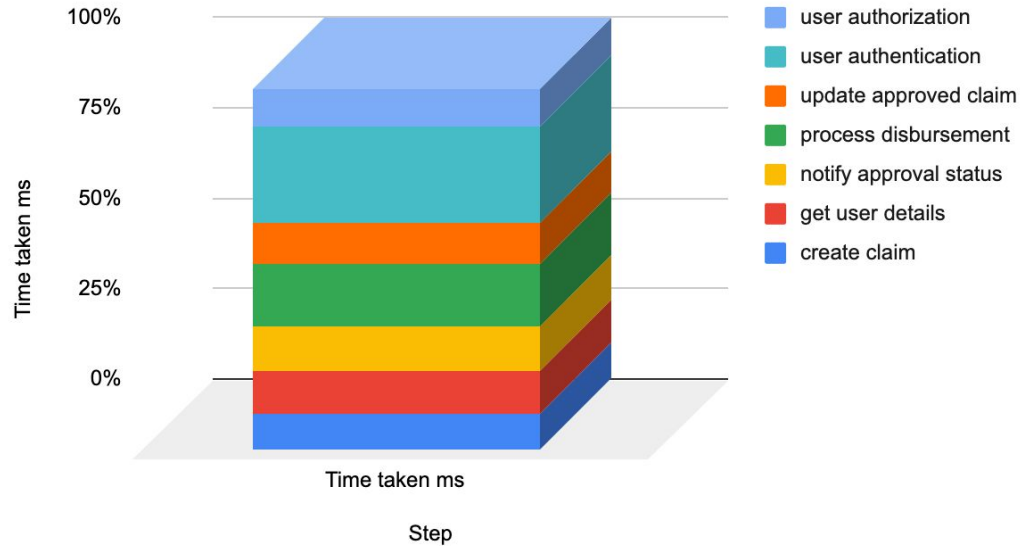
DEMO WORKFLOW

Live Demo

DEMO

Statistics, metrics and dashboards

Distribution of Time taken ms by Step



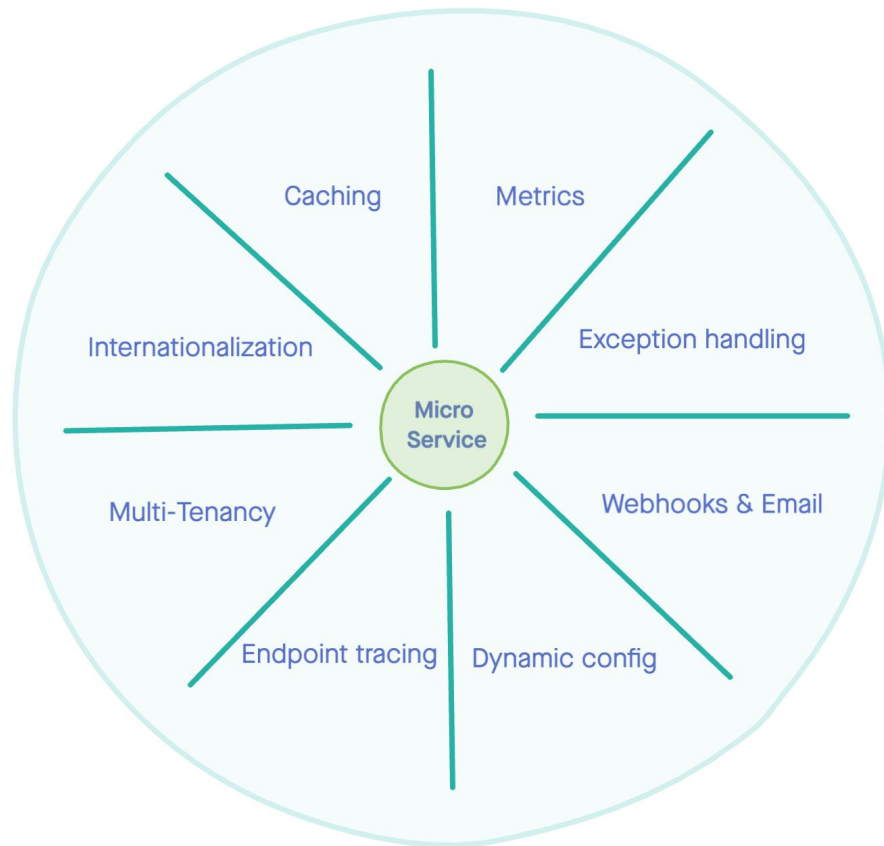
CRUX - A microservices kernel library

Bootstrap consistent and feature-rich microservices and focus on business value delivery

Typical challenges

Challenges in microservices architecture

- Lack of consistency means each service is novel in implementation.
- Lack of standards mean higher effort on productionizing, operationalizing. Low on observability and coherence.
- Capabilities like caching, l18n, Metrics, environment config management and multi-tenancy cost money and time.
- Cross functional teams can suffer from diversity fatigue or there can be strong service affinity in large distributed systems



Batteries included

Scaffolding consistent and architecturally robust microservices

Endpoint decoration

Annotate and mark endpoints with Crux. Automatically collect and visualize endpoint usage data as well as the response times.

Caching

Simple in-memory caches out of the box. Low cost implementation to keep services nimble and swift in response times.

I18n

A sensible approach to managing internationalization in info and error messages. Implemented out of the box with message placeholders.

Metrics for everything

Crux provides metrics for endpoints usage, cache efficacy, configs and lot more things. In addition to that it exposes metrics registry to stream data to preferred metrics store.

Webhook, Slack and Email integrations

Framework provides out of the box support for webhooks, slack integration as well as email integrations.

Multi-tenancy

Multi-tenancy, a feature that requires significant development cycles to get right, comes out of the box with smart tenant-based property management

Dynamic env configs

View and manage environment configs dynamically using a git backed config server. It allows for auditability and effective debugging remote environments.

Standardized exception handling

Simple, stateless and scalable structure backed by a persistent store. The orchestrator can scale and manage high throughput with ease with minimal overhead.

Live Demo

DEMO

Reference slides and additional information

Step 1 - Authentication

```
{
  "claimDetails": {
    "claimDate": "23/06/2020",
    "claimType": "MOTOR",
    "garageId": "5000",
    "policyId": "908",
    "supportingDocuments": ""
  },
  "userDetails": {
    "channel": "web",
    "password": "jungle-green-t0p!",
    "username": "shambhu.shikari"
  }
}
```

Request

```
"response": {
  "id": "1234567890",
  "name": "Shambhu Shikari",
  "username": "shambhu.shikari"
}
```

Response

Step 2 - Authorization

```
{  
  "userId": "1234567890",  
  "username": "shambhu.shikari"  
}
```

Request

```
{  
  "roles": [  
    "RETAIL_CUSTOMER"  
  ],  
  "userId": "1234567890",  
  "username": "shambhu.shikari"  
}
```

Response

Step 3 - User details

```
{
  "roles": [
    "RETAIL_CUSTOMER"
  ],
  "userId": "1234567890",
  "username": "shambhu.shikari"
}
```

Request

```
{
  "city": "Bangalore",
  "emailId": "shambhu.shikari@maildrop.cc",
  "mobileNumber": "990099009900",
  "policyDetails": [
    {
      "policyId": "908",
      "policyType": "MOTOR",
      "validityFrom": "2020 Apr 13 00:00:00.000 IST",
      "validityTo": "2021 Apr 12 00:00:00.000 IST"
    },
    {
      "policyId": "610",
      "policyType": "MEDICAL",
      "validityFrom": "2019 Aug 13 00:00:00.000 IST",
      "validityTo": "2020 Aug 12 00:00:00.000 IST"
    }
  ],
  "state": "karnataka",
  "userId": "1234567890",
  "username": "shambhu.shikari"
}
```

Response

Step 4 - Create claim

```
{
  "claimDetails": {
    "claimDate": "23/06/2020",
    "claimType": "MOTOR",
    "garageId": "5000",
    "policyId": "908",
    "supportingDocuments": ""
  },
  "existingPolicies": [
    {
      "policyId": "908",
      "policyType": "MOTOR",
      "validityFrom": "2020 Apr 13 00:00:00.000 IST",
      "validityTo": "2021 Apr 12 00:00:00.000 IST"
    },
    {
      "policyId": "610",
      "policyType": "MEDICAL",
      "validityFrom": "2019 Aug 13 00:00:00.000 IST",
      "validityTo": "2020 Aug 12 00:00:00.000 IST"
    }
  ],
  "userId": "1234567890"
}
```

Request

```
{
  "claimDate": "2020 Jun 23 00:00:00.000 IST",
  "claimId": "90990908324",
  "claimStatus": "INITIATED",
  "claimType": "MOTOR",
  "garageId": "5000",
  "inspectorDetails": {
    "assignee": "Shivu",
    "assigneeId": "12924",
    "inspectionDate": "2020 Jun 23 00:00:00.000 IST"
  },
  "notes": "",
  "policyId": "908",
  "reviewerDetails": {},
  "userId": "1234567890"
}
```

Response

Step 5 - Submit claim - *Async*

```
{
  "claimDate": "2020 Jun 23 00:00:00.000 IST",
  "claimId": "90990908324",
  "claimStatus": "INITIATED",
  "claimType": "MOTOR",
  "garageId": "5000",
  "inspectorDetails": {
    "assignee": "Shivu",
    "assigneeId": "12924",
    "inspectionDate": "2020 Jun 23 00:00:00.000 IST"
  },
  "policyId": "908",
  "userId": "1234567890"
}
```

Request

Step 6 - Approve claim - *Manual action*

```
{
  "claimDate": "23/06/2020",
  "claimId": "90990908324",
  "claimStatus": "APPROVED",
  "claimType": "MOTOR",
  "garageId": "5000",
  "notes": "Inspection not required approved based on documentation. CASHLESS",
  "policyId": "908",
  "reviewerDetails": {
    "approvedAmount": "5000",
    "reviewDate": "23/06/2020",
    "reviewerId": "12924",
    "reviewerName": "Sanjeev"
  },
  "userId": "1234567890"
}
```

Response

Step 7 - Update claim

```
{
  "claimDate": "2020 Jun 23 00:00:00.000 IST",
  "claimId": "90990908324",
  "claimStatus": "APPROVED",
  "claimType": "MOTOR",
  "garageId": "5000",
  "inspectorDetails": {
    "assignee": "Shivu",
    "assigneeId": "12924",
    "inspectionDate": "2020 Jun 23 00:00:00.000 IST"
  },
  "policyId": "908",
  "userId": "1234567890"
}
```

Request

```
{
  "claimDate": "2020 Jun 23 00:00:00.000 IST",
  "claimId": "90990908324",
  "claimStatus": "APPROVED",
  "claimType": "MOTOR",
  "garageId": "5000",
  "inspectorDetails": {
    "assignee": "Shivu",
    "assigneeId": "12924",
    "inspectionDate": "2020 Jun 23 00:00:00.000 IST"
  },
  "notes": "Inspection not required approved based on documentation. CASHLESS",
  "policyId": "908",
  "reviewerDetails": {
    "approvedAmount": "5000",
    "reviewDate": "2020 Jun 23 00:00:00.000 IST",
    "reviewerId": "12924",
    "reviewerName": "Sanjeev"
  },
  "userId": "1234567890"
}
```

Response

Step 8 - Initiate disbursement

```
{
  "approvedAmount": "5000",
  "claimId": "90990908324",
  "claimStatus": "APPROVED",
  "reviewerDate": "2020 Jun 23 00:00:00.000 IST",
  "reviewerId": "12924",
  "userId": "1234567890"
}
```

Request

```
{
  "claimId": "90990908324",
  "disbursedAmount": "5000",
  "disbursementDate": "2020 Jun 23 00:00:00.000 IST",
  "disbursementRefId": "234234434",
  "partyDetails": {
    "partyId": "23432431",
    "partyName": "Apple Auto"
  },
  "paymentInstrumentId": "CHEQUE",
  "userId": "1234567890"
}
```

Response

Step 9 - Send notification

```
{
  "claimId": "90990908324",
  "claimStatus": "APPROVED",
  "claimType": "MOTOR",
  "disbursedAmount": "5000",
  "disbursementDate": "2020 Jun 23 00:00:00.000 IST",
  "emailId": "shambhu.shikari@maildrop.cc",
  "mobileNumber": "990099009900",
  "partyName": "Apple Auto",
  "reviewerName": "Sanjeev",
  "userId": "1234567890",
  "username": "shambhu.shikari"
}
```

Request

```
{
  "notifications": [
    {
      "notificationAddresss": "shambhu.shikari@maildrop.cc",
      "notificationContent": "Dear Shambhu Shikari, Your MOTOR claim",
      "notificationDate": "23/06/2020",
      "notificationStatus": "SUCCESS",
      "notificationType": "EMAIL"
    }
  ]
}
```

Response



Bengaluru
Chennai
Pune
London
Melbourne
SF Bay Area

www.sahajsoft.com
connect@sahajsoft.com
[@SahajSoftware](https://twitter.com/SahajSoftware)

For information and discussion purposes only, and not to be binding document.

Only for the use of intended recipients. Do not share without prior permission.

©2020 Sahaj Software.