Eclipse Muto

An adaptive ROS **orchestration framework** for dynamic and model-driven software stacks

Naci Dai, Deniz Memiş
naci.dai,deniz.memis@composiv.ai

Speakers

Naci Dai

naci@composiv.ai

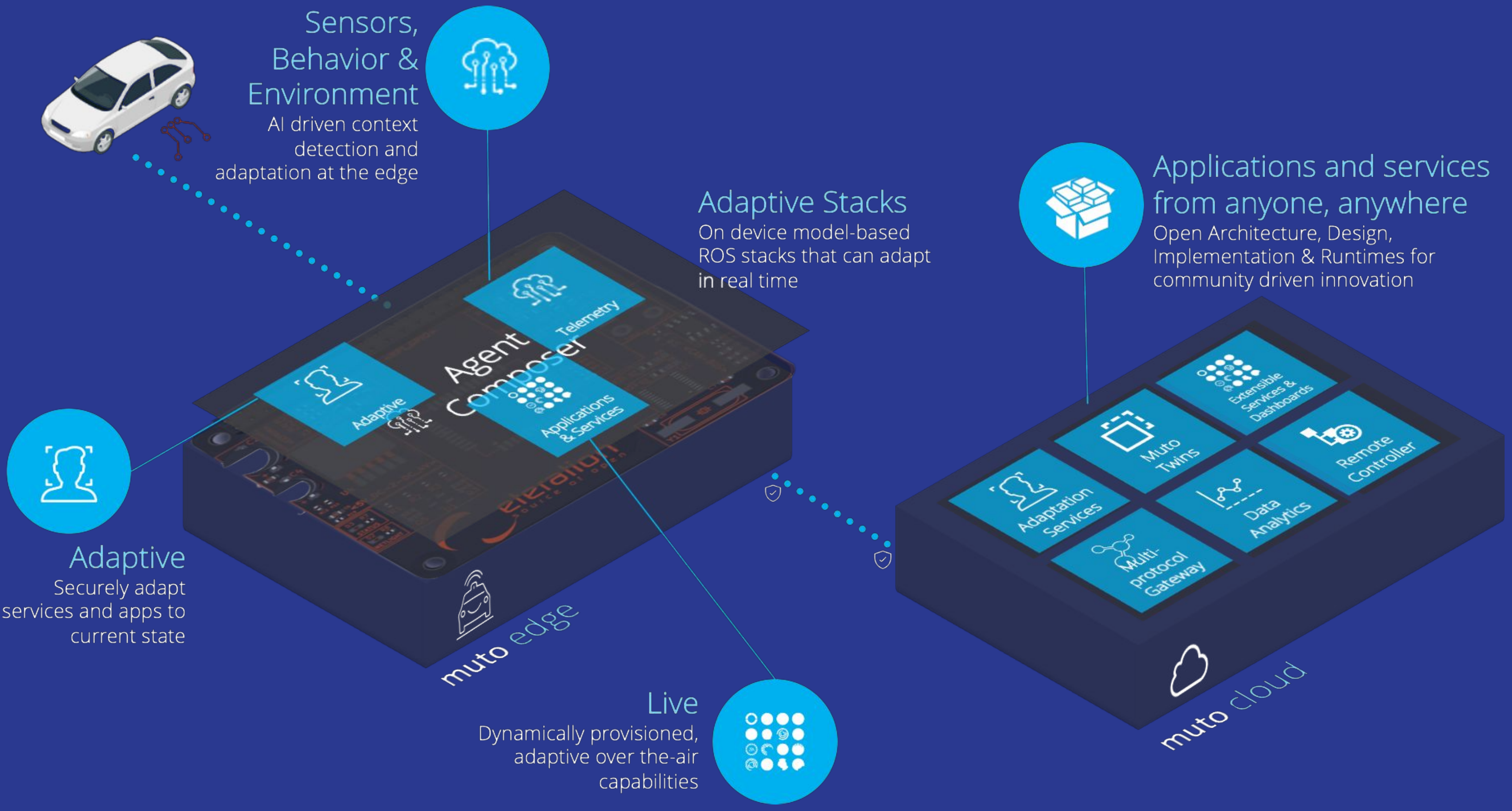
Deniz Memiş

deniz@composiv.ai

composiv.ai

Adaptive Solutions
Context Aware Adaptive
Software Stacks for Mobility

muto



Sensors, Behavior & Environment

AI driven context detection and adaptation at the edge

Adaptive Stacks

On device model-based ROS stacks that can adapt in real time

Applications and services from anyone, anywhere

Open Architecture, Design, Implementation & Runtimes for community driven innovation

Adaptive

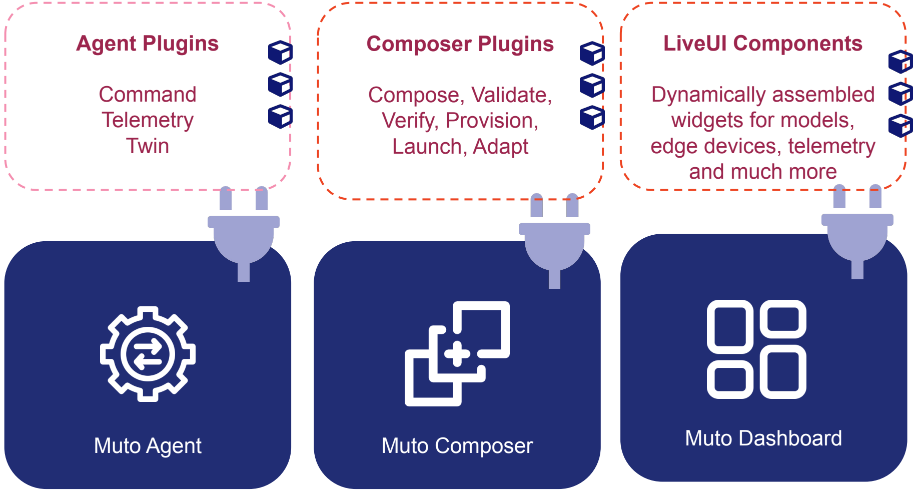
Securely adapt services and apps to current state

Live

Dynamically provisioned, adaptive over the-air capabilities

eclipse muto

an adaptive framework and a runtime platform for dynamically composable model-driven software stacks for ROS



Agent Plugins

Command
Telemetry
Twin

Composer Plugins

Compose, Validate,
Verify, Provision,
Launch, Adapt

LiveUI Components

Dynamically assembled
widgets for models,
edge devices, telemetry
and much more

Muto Agent

Muto Composer

Muto Dashboard

Challenges

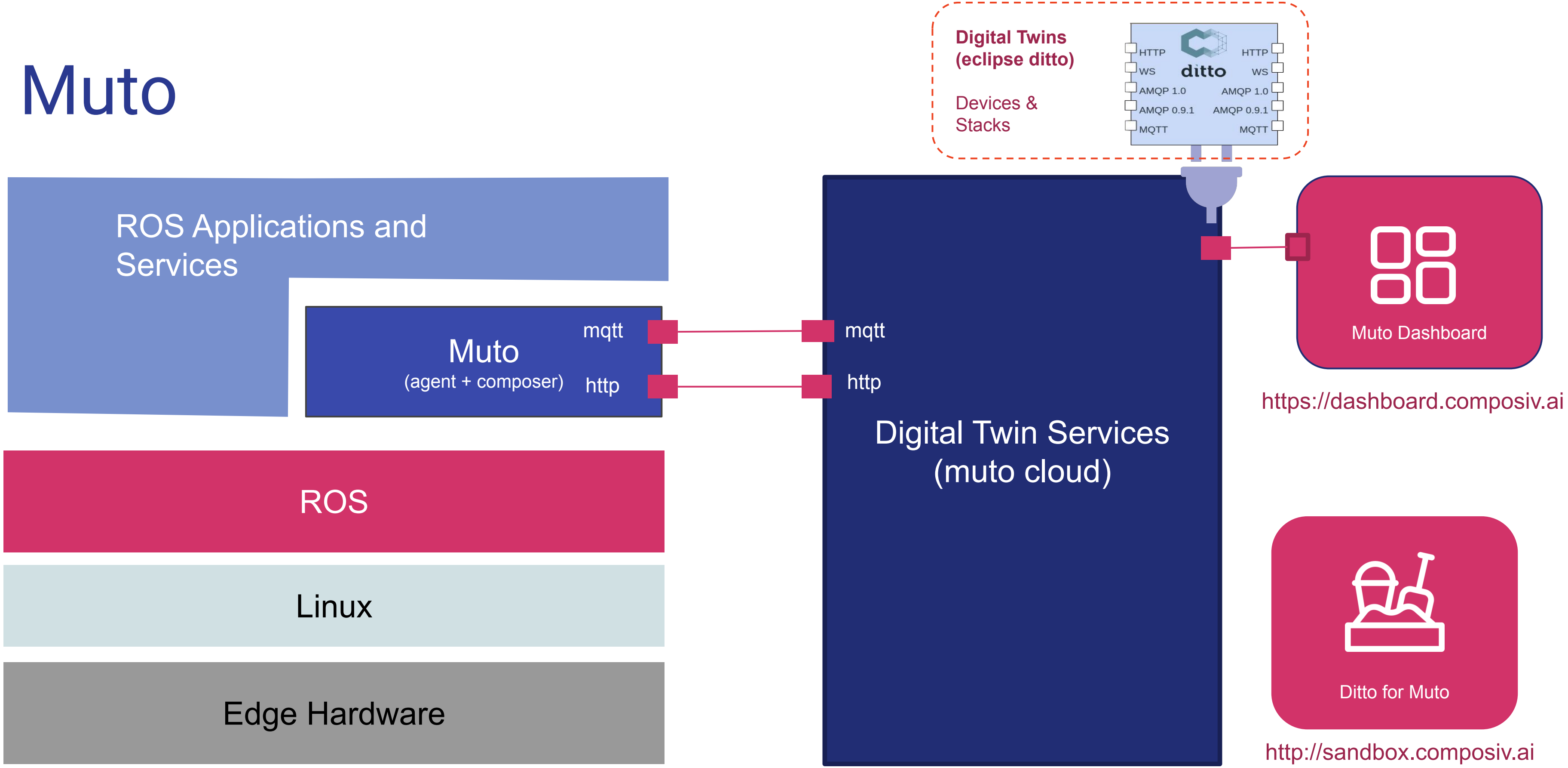
Bottom up
Closed & proprietary
Builtin biases
Hardwired & static
Built for a purpose

Response

Model driven
Open
Contextual
Dynamic & Live
Adaptive & Extensible

icons made by bqlqn from flaticon.com
Icons made by freepik from flaticon.com

Muto



Digital Twins (eclipse ditto)

Devices & Stacks

ROS Applications and Services

Muto (agent + composer)

mqtt

http

ROS

Linux

Edge Hardware

mqtt

http

Digital Twin Services (muto cloud)

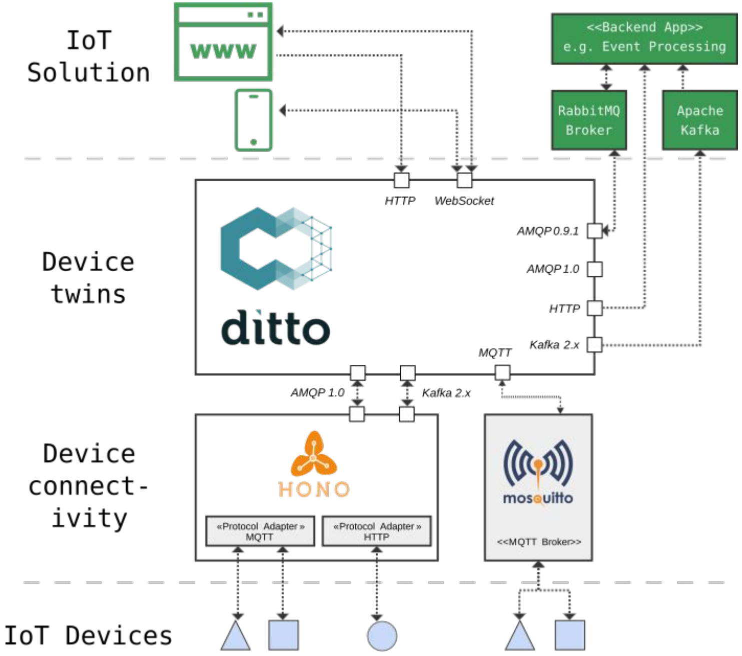
Muto Dashboard

https://dashboard.composiv.ai

Ditto for Muto

http://sandbox.composiv.ai

muto uses ditto for stack and vehicle digital twins*



This diagram is reproduced from https://www.eclipse.org/ditto

stack

vehicle

*A digital twin is a virtual representation that serves as the real-time digital counterpart of a physical object or a process.

icons made by bqlqn from flaticon.com
Icons made by freepik from flaticon.com

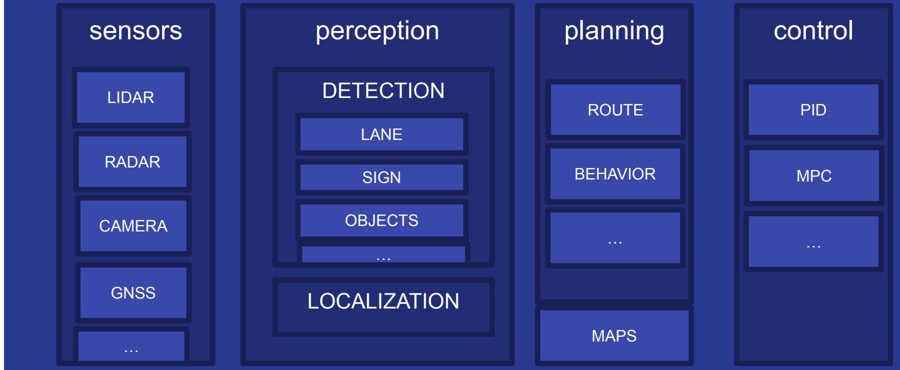
muto

Stack

A dynamic model of ROS software running on a device

“a deployment manifest”

By Example

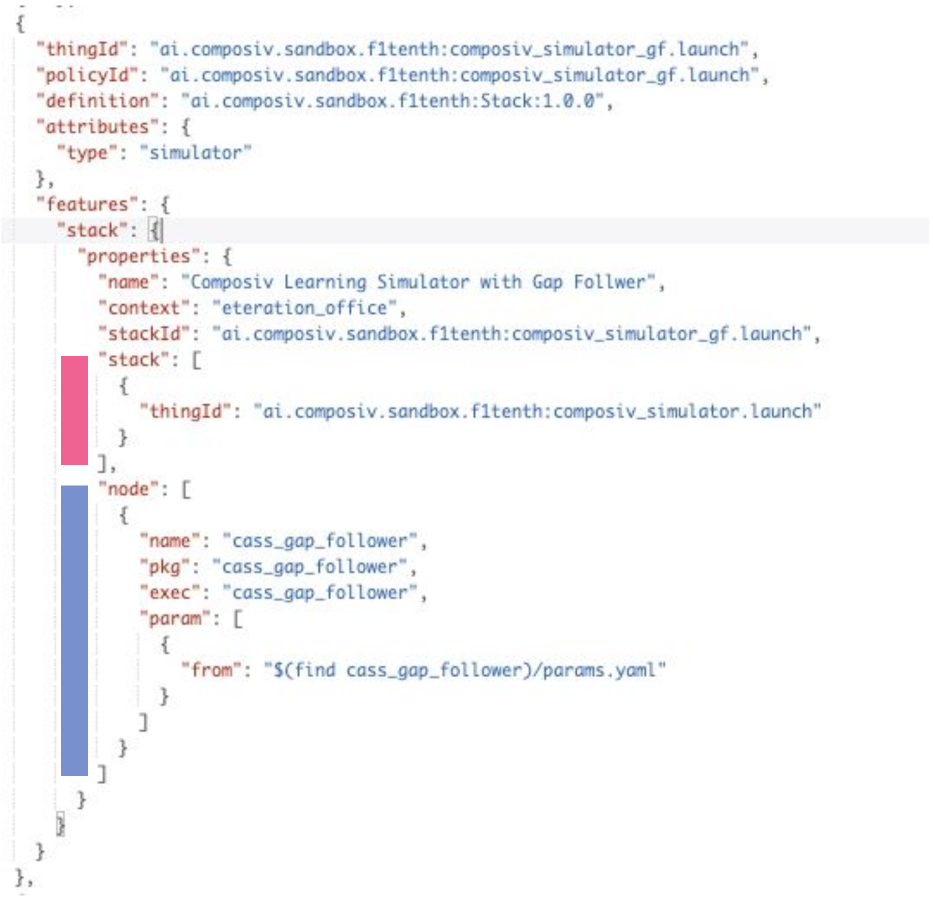


```
{
"name": "Muto Example Stack",
"stackId": ”muto:example",
"arg": [ … ],
"param": […],
”stack": […],
"node": […]
}
```

Stack

Stack Model

- Follows ROS constructs
  - args/params/nodes/topics/…
- Modular
  - reference other stack “things”



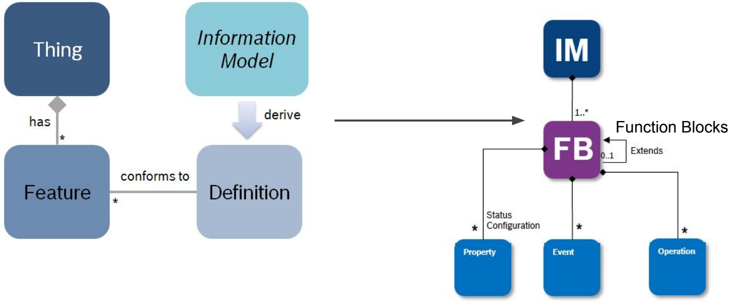
```
{
  "thingId": "ai.composiv.sandbox.f1tenth:composiv_simulator_gf.launch",
  "policyId": "ai.composiv.sandbox.f1tenth:composiv_simulator_gf.launch",
  "definition": "ai.composiv.sandbox.f1tenth:Stack:1.0.0",
  "attributes": {
    "type": "simulator"
  },
  "features": {
    "stack": {
      "properties": {
        "name": "Composiv Learning Simulator with Gap Follwer",
        "context": "eteration_office",
        "stackId": "ai.composiv.sandbox.f1tenth:composiv_simulator_gf.launch",
        "stack": [
          {
            "thingId": "ai.composiv.sandbox.f1tenth:composiv_simulator.launch"
          }
        ],
        "node": [
          {
            "name": "cass_gap_follower",
            "pkg": "cass_gap_follower",
            "exec": "cass_gap_follower",
            "param": [
              {
                "from": "$(find cass_gap_follower)/params.yaml"
              }
            ]
          }
        ]
      }
    }
  }
},
```

Vehicle

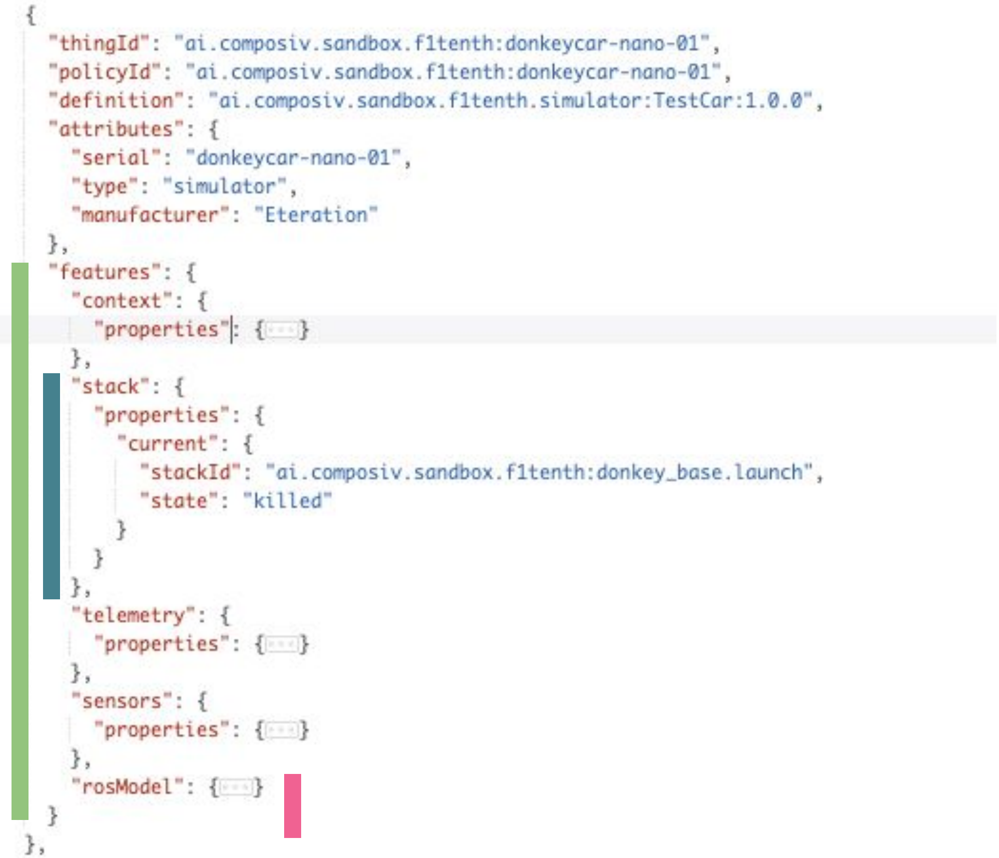
Associate a stack with a vehicle and manage its lifecycle

ROS State (nodes/topics/params)

Telemetry and sensor metadata

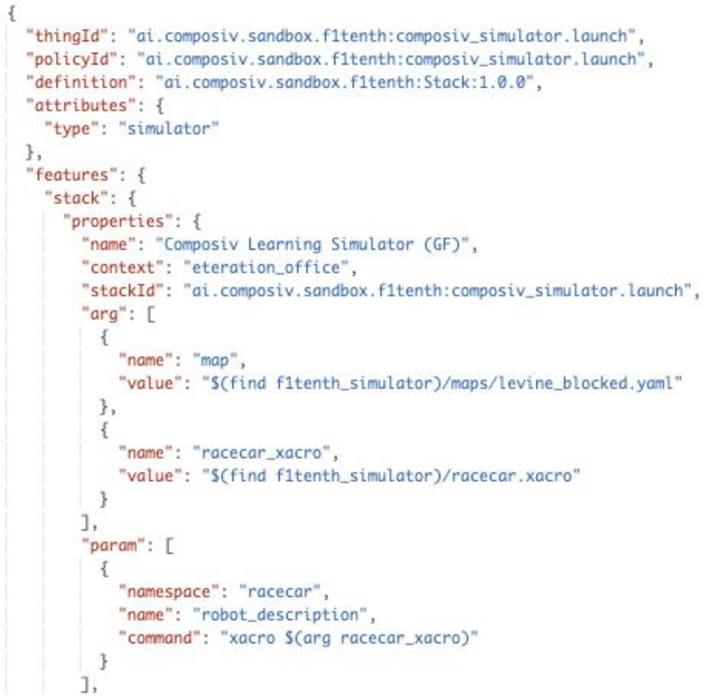


Class diagram for things from https://www.eclipse.org/ditto

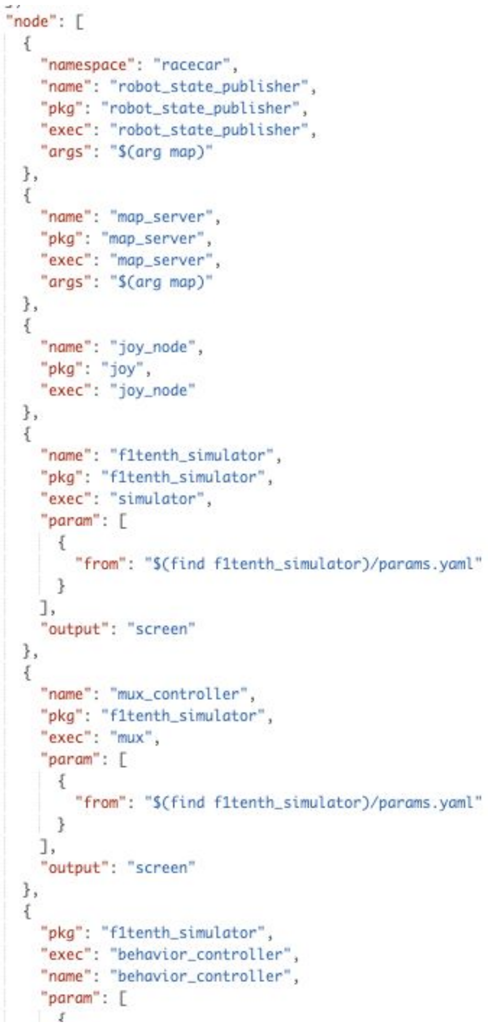


```
{
  "thingId": "ai.composiv.sandbox.f1tenth:donkeycar-nano-01",
  "policyId": "ai.composiv.sandbox.f1tenth:donkeycar-nano-01",
  "definition": "ai.composiv.sandbox.f1tenth.simulator:TestCar:1.0.0",
  "attributes": {
    "serial": "donkeycar-nano-01",
    "type": "simulator",
    "manufacturer": "Eteration"
  },
  "features": {
    "context": {
      "properties": {…}
    },
    "stack": {
      "properties": {
        "current": {
          "stackId": "ai.composiv.sandbox.f1tenth:donkey_base.launch",
          "state": "killed"
        }
      }
    },
    "telemetry": {
      "properties": {…}
    },
    "sensors": {
      "properties": {…}
    },
    "rosModel": {…}
  }
},
```

more Stack …



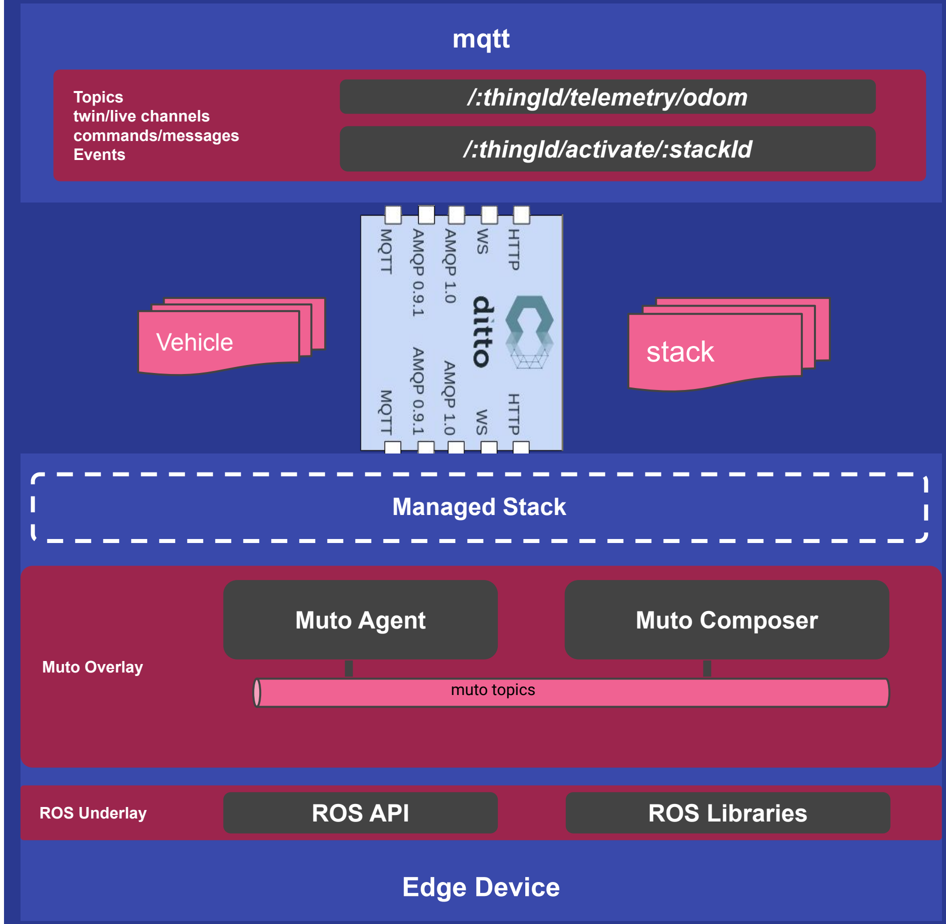
```
{
  "thingId": "ai.composiv.sandbox.f1tenth:composiv_simulator.launch",
  "policyId": "ai.composiv.sandbox.f1tenth:composiv_simulator.launch",
  "definition": "ai.composiv.sandbox.f1tenth:Stack:1.0.0",
  "attributes": {
    "type": "simulator"
  },
  "features": {
    "stack": {
      "properties": {
        "name": "Composiv Learning Simulator (GF)",
        "context": "eteration_office",
        "stackId": "ai.composiv.sandbox.f1tenth:composiv_simulator.launch",
        "arg": [
          {
            "name": "map",
            "value": "$(find f1tenth_simulator)/maps/levine_blocked.yaml"
          },
          {
            "name": "racecar_xacro",
            "value": "$(find f1tenth_simulator)/racecar.xacro"
          }
        ],
        "param": [
          {
            "namespace": "racecar",
            "name": "robot_description",
            "command": "xacro $(arg racecar_xacro)"
          }
        ],
```



```
"node": [
  {
    "namespace": "racecar",
    "name": "robot_state_publisher",
    "pkg": "robot_state_publisher",
    "exec": "robot_state_publisher",
    "args": "$(arg map)"
  },
  {
    "name": "map_server",
    "pkg": "map_server",
    "exec": "map_server",
    "args": "$(arg map)"
  },
  {
    "name": "joy_node",
    "pkg": "joy",
    "exec": "joy_node"
  },
  {
    "name": "f1tenth_simulator",
    "pkg": "f1tenth_simulator",
    "exec": "simulator",
    "param": [
      {
        "from": "$(find f1tenth_simulator)/params.yaml"
      }
    ],
    "output": "screen"
  },
  {
    "name": "mux_controller",
    "pkg": "f1tenth_simulator",
    "exec": "mux",
    "param": [
      {
        "from": "$(find f1tenth_simulator)/params.yaml"
      }
    ],
    "output": "screen"
  },
  {
    "pkg": "f1tenth_simulator",
    "exec": "behavior_controller",
    "name": "behavior_controller",
    "param": [
      {
```

Agent

- Edge Gateway
  - A runtime ROS component
  - Device authentication and activation
- Plugins
  - Commands: Stack Management
  - Commands: ROS
  - Commands: Telemetry
- Asynchronous connection protocol
  - relay messages to *plugins*
  - Telemetry Streams



mqtt

Topics
twin/live channels
commands/messages
Events

/:thingId/telemetry/odom

/:thingId/activate/:stackId

Vehicle

stack

Managed Stack

Muto Overlay

Muto Agent

Muto Composer

muto topics

ROS Underlay

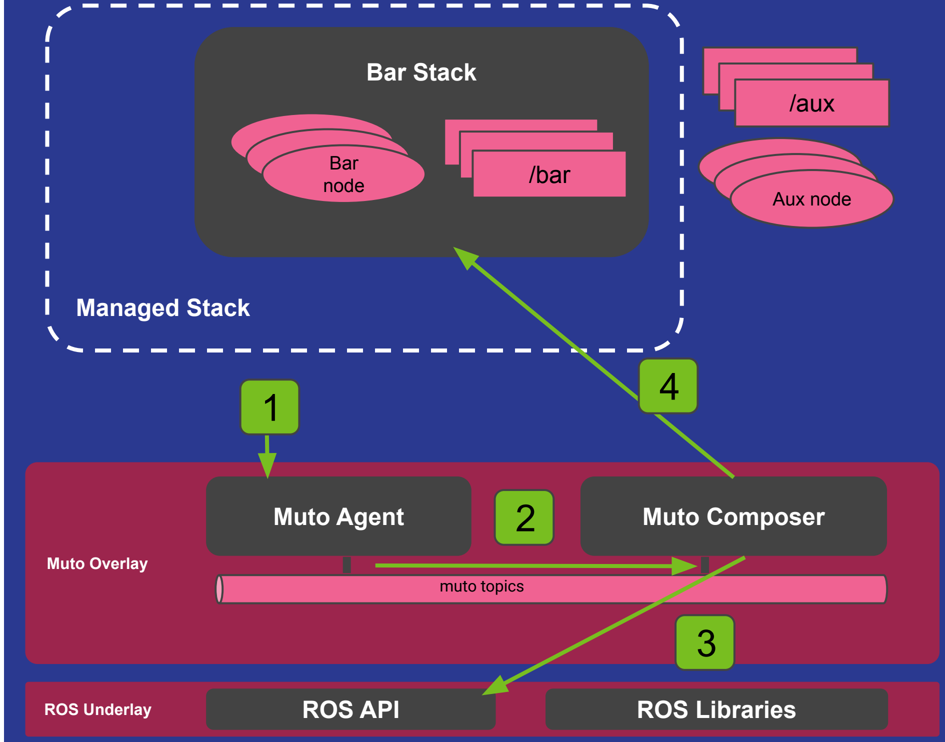
ROS API

ROS Libraries

Edge Device

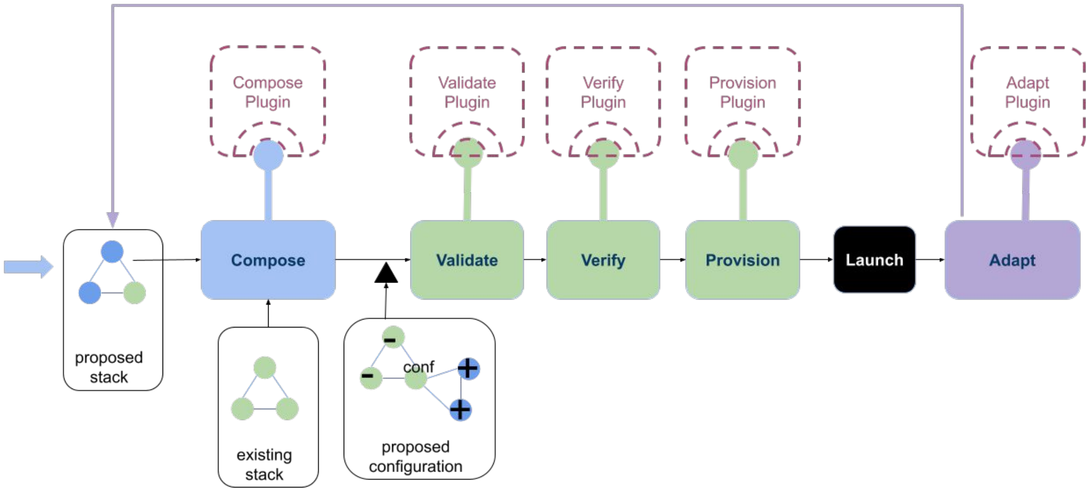
Composer

- Orchestration middleware
  - Workflow
  - Adapt stack (models)
  - Component lifecycle
- Plugins
  - Compose: ROS Graph Algebra
  - Compose: Validate
  - Compose: Verify
  - Compose: Provision
  - Compose: Launch
  - Compose: Adapt



muto

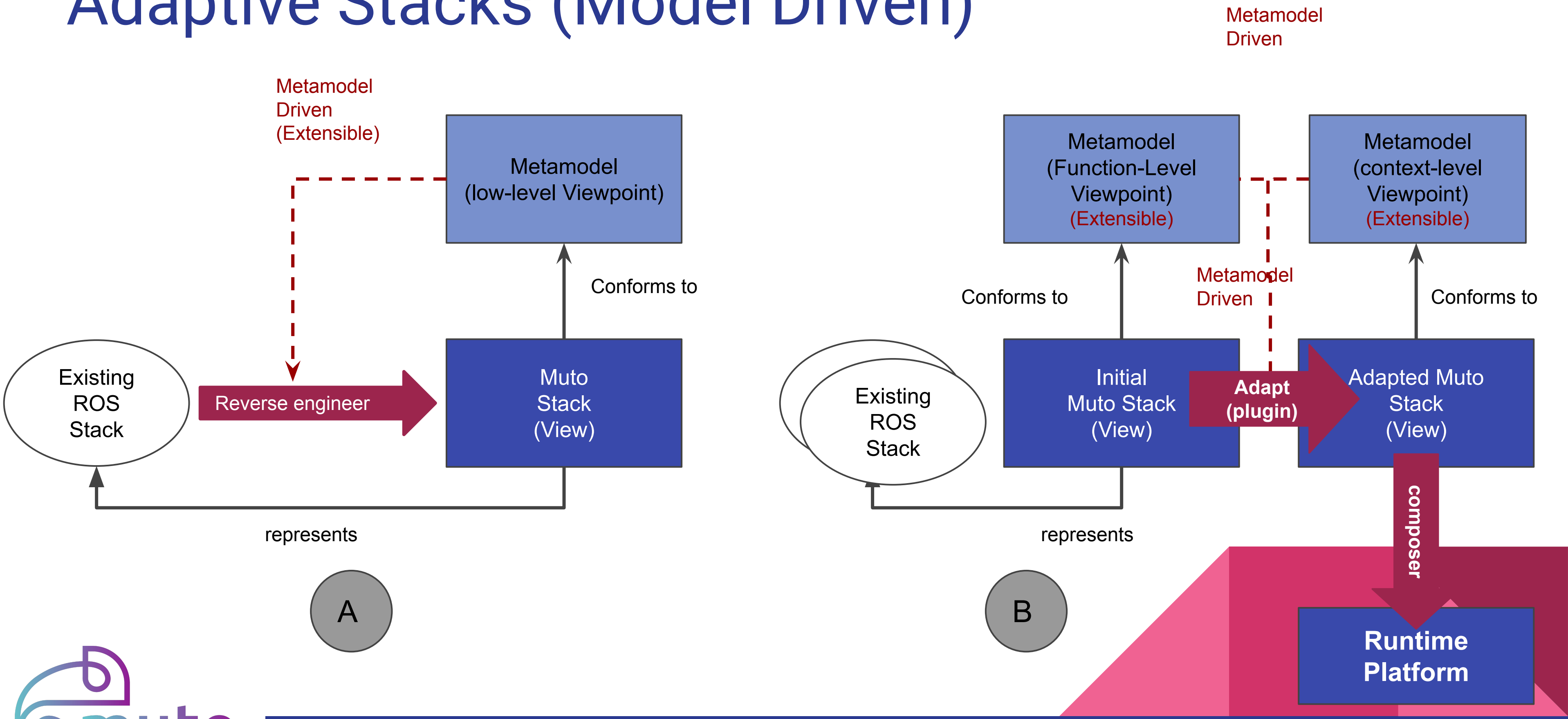
Extensible Dynamic Workflows



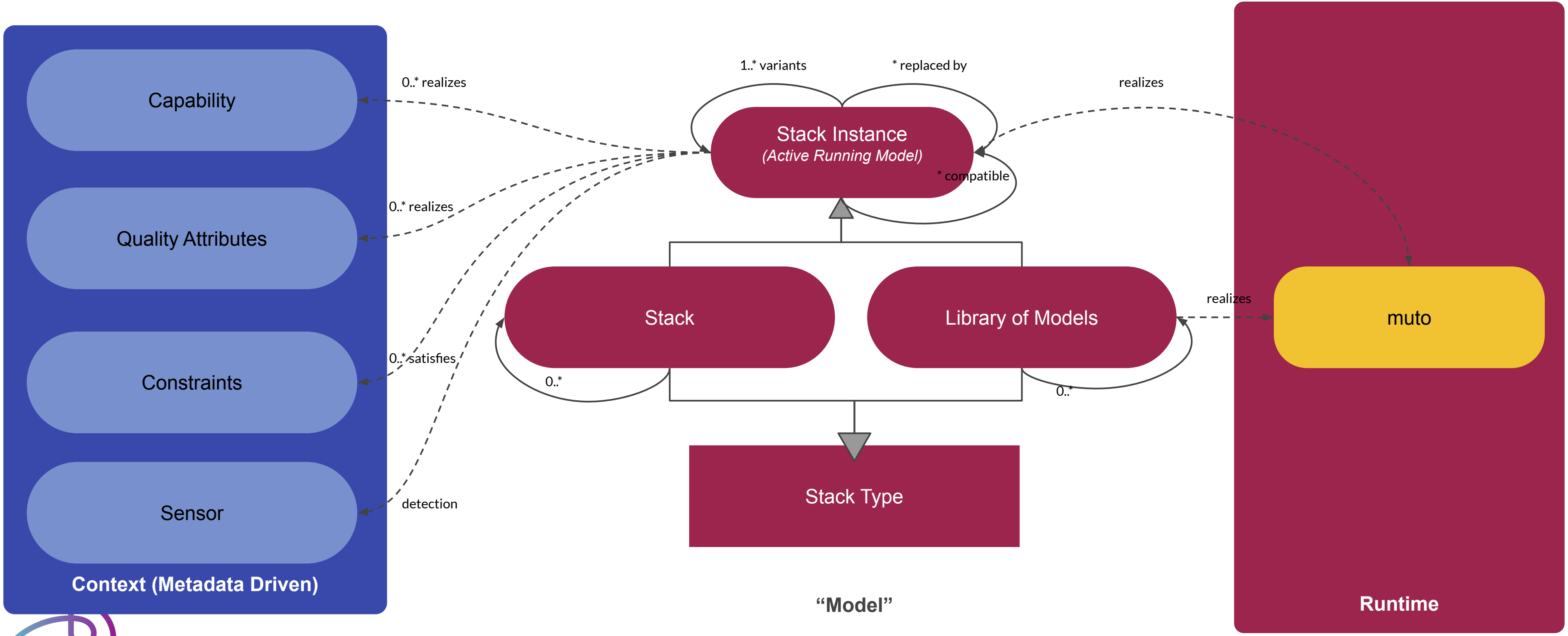
```
flow:
  - name:  start
    pipeline:
      - service: muto_compose
        plugin: ComposePlugin
      - service: muto_validate
        plugin: ComposePlugin
      - service: muto_verify
        plugin: ComposePlugin
      - service: muto_provision
        plugin: ComposePlugin
      - service: muto_start_stack
        plugin: ComposePlugin
      - service: muto_adapt
        plugin: ComposePlugin
    compensation:
      - service: muto_kill_stack
        plugin: ComposePlugin

  - name:  stop
    pipeline:
      ..
```

Muto Plugins

Adaptive Stacks (Model Driven)

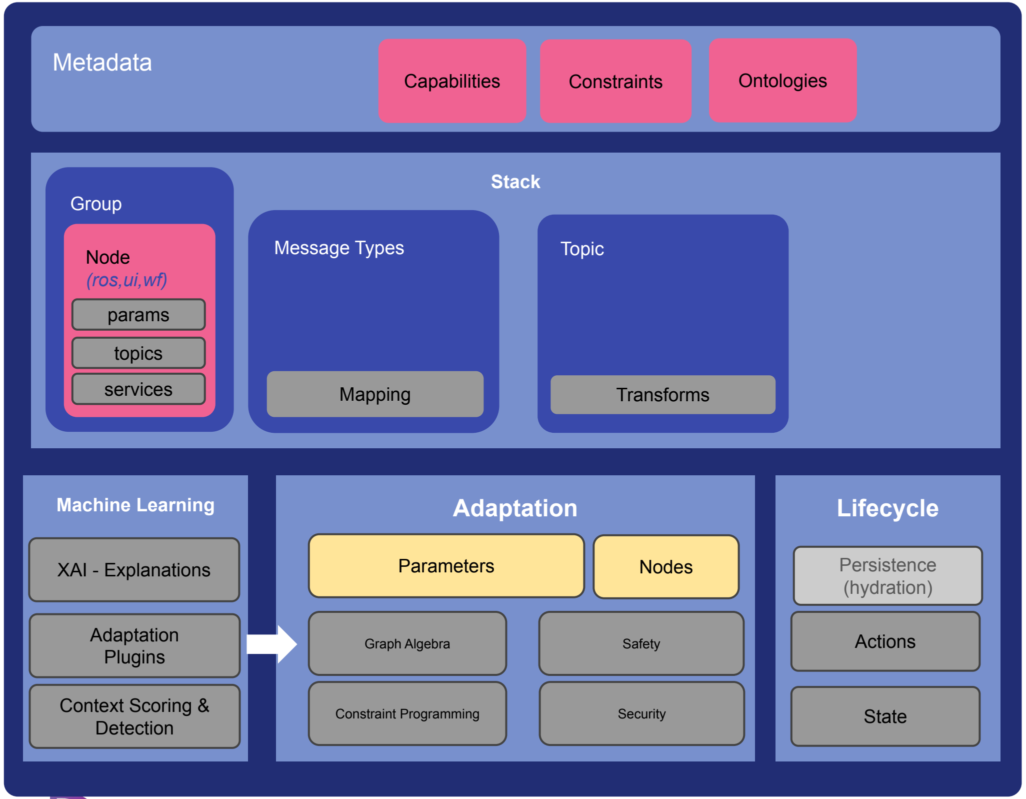


Contextual Adaptation



muto

Overview



Metadata

Capabilities

Constraints

Ontologies

Stack

Group

Node

(ros,ui,wf)

params

topics

services

Message Types

Mapping

Topic

Transforms

Machine Learning

XAI - Explanations

Adaptation Plugins

Context Scoring & Detection

Adaptation

Parameters

Nodes

Graph Algebra

Safety

Constraint Programming

Security

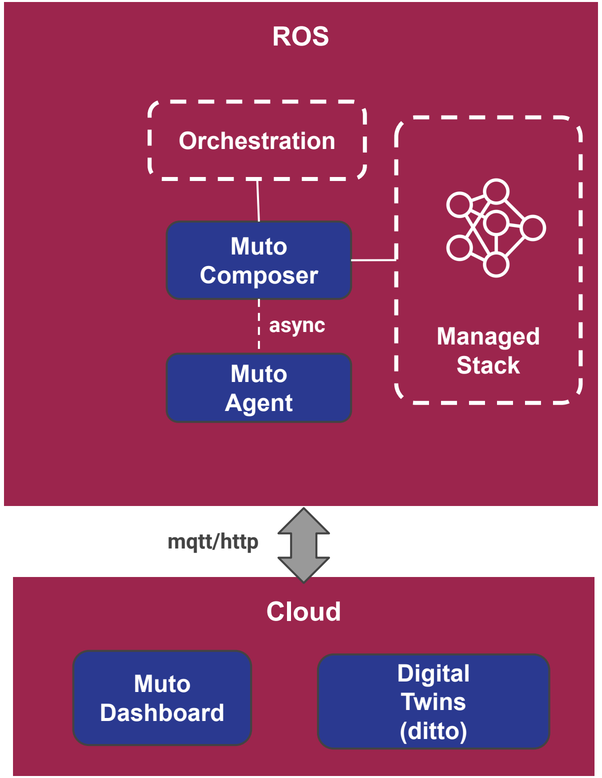
Lifecycle

Persistence (hydration)

Actions

State

muto runtimes



ROS

Orchestration

Muto Composer

async

Muto Agent

Managed Stack

mqtt/http

Cloud

Muto Dashboard

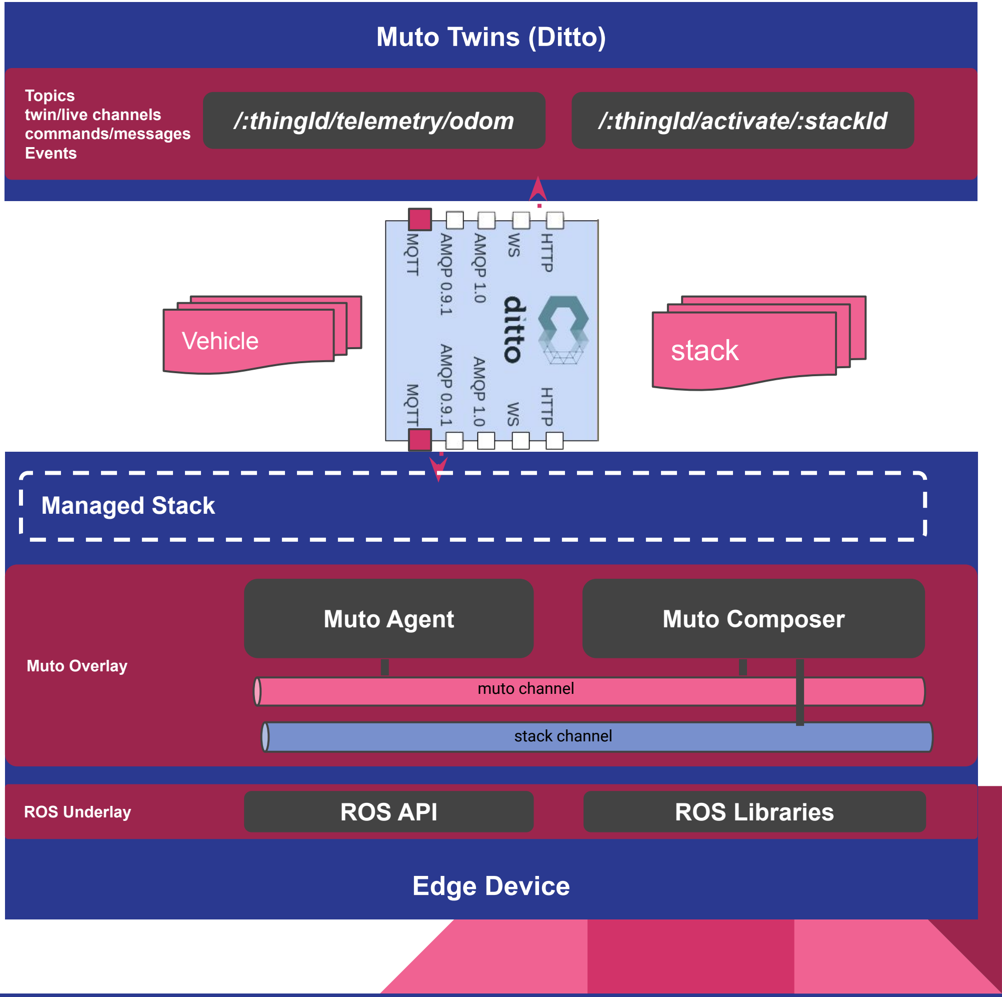
Digital Twins (ditto)

muto

icons made by bqlqn from flaticon.com
Icons made by freepik from flaticon.com

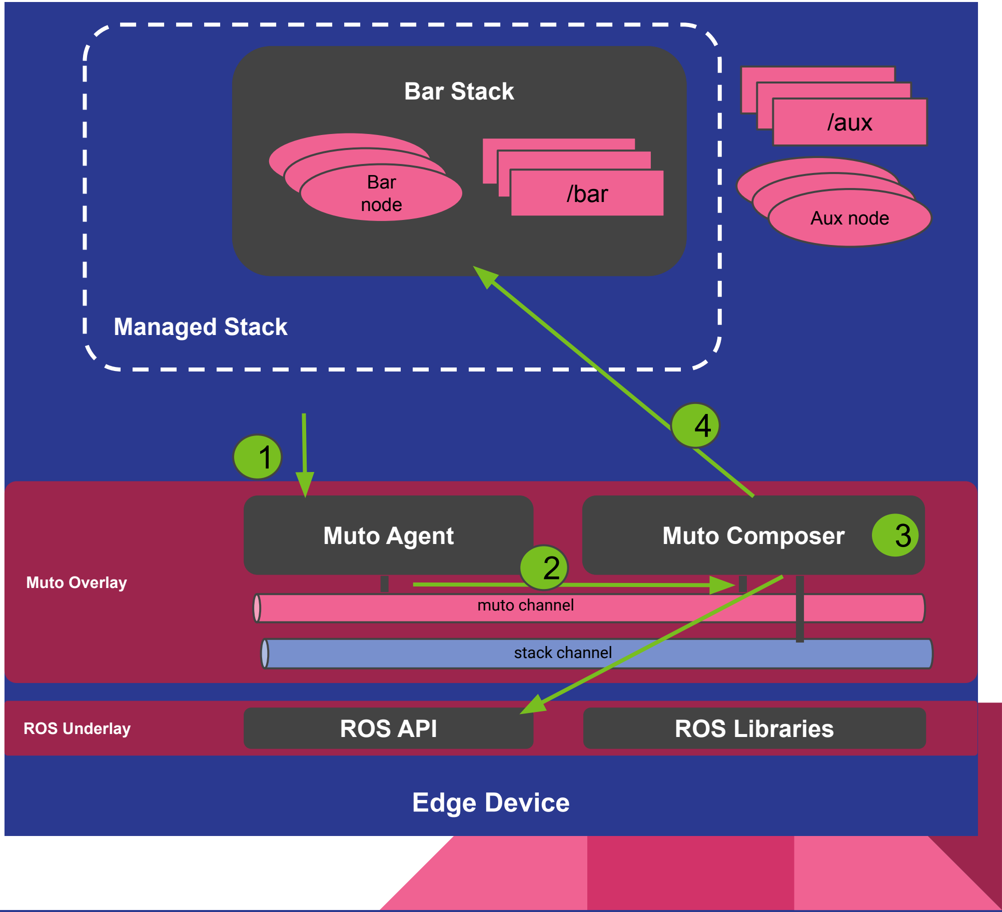
Muto Agent

- A runtime ROS component
  - Vehicle(device) authentication and activation
- Gateway to edge
  - remote management capabilities
    - i.e. eclipse ditto twins
  - Cloud to ROS
- Bidirectional (cloud <-> edge)
  - Relays messages to *Composer* for stack lifecycle management
  - Streams edge device information
- Asynchronous



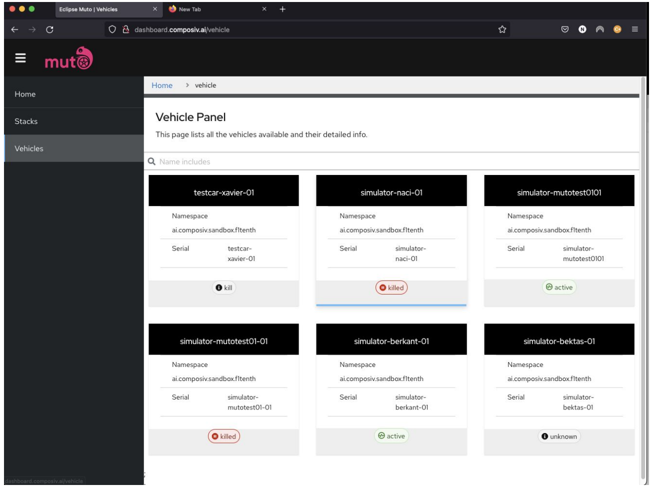
Muto Composer

- A runtime ROS component
- Composes and manage
  - life cycle of ROS nodes
  - defined by the Stack (model)
- Node graph algebra
  - Stack introspection
  - Stack diff
  - Node Lifecycle actions
  - Interact with Param server



Muto Dashboard

- Centralized management
  - Extensible and modular
  - µFrontends
  - Muto LiveUI
- Exemplars
  - ROS Inspection
  - Vehicles
  - Stack & Lifecycle Management
  - Telemetry



Vehicle Panel

This page lists all the vehicles available and their detailed info.

Name includes

testcar-xavier-01
Namespace ai.composiv.sandbox.f1tenth
Serial testcar-xavier-01
kill

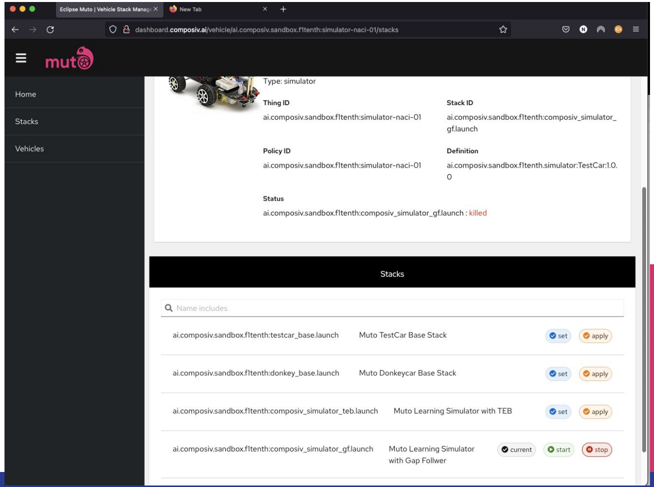
simulator-naci-01
Namespace ai.composiv.sandbox.f1tenth
Serial simulator-naci-01
killed

simulator-mutotest0101
Namespace ai.composiv.sandbox.f1tenth
Serial simulator-mutotest0101
active

simulator-mutotest01-01
Namespace ai.composiv.sandbox.f1tenth
Serial simulator-mutotest01-01
killed

simulator-berkant-01
Namespace ai.composiv.sandbox.f1tenth
Serial simulator-berkant-01
active

simulator-bektas-01
Namespace ai.composiv.sandbox.f1tenth
Serial simulator-bektas-01
unknown



Eclipse Muto | Vehicle Stack Manag
New Tab
dashboard.composiv.ai/vehicle/ai.composiv.sandbox.f1tenth:simulator-naci-01/stacks
muto
Home
Stacks
Vehicles

Type: simulator

Thing ID
ai.composiv.sandbox.f1tenth:simulator-naci-01

Stack ID
ai.composiv.sandbox.f1tenth:composiv_simulator_gf.launch

Policy ID
ai.composiv.sandbox.f1tenth:simulator-naci-01

Definition
ai.composiv.sandbox.f1tenth.simulator:TestCar:1.0.0

Status
ai.composiv.sandbox.f1tenth:composiv_simulator_gf.launch : killed

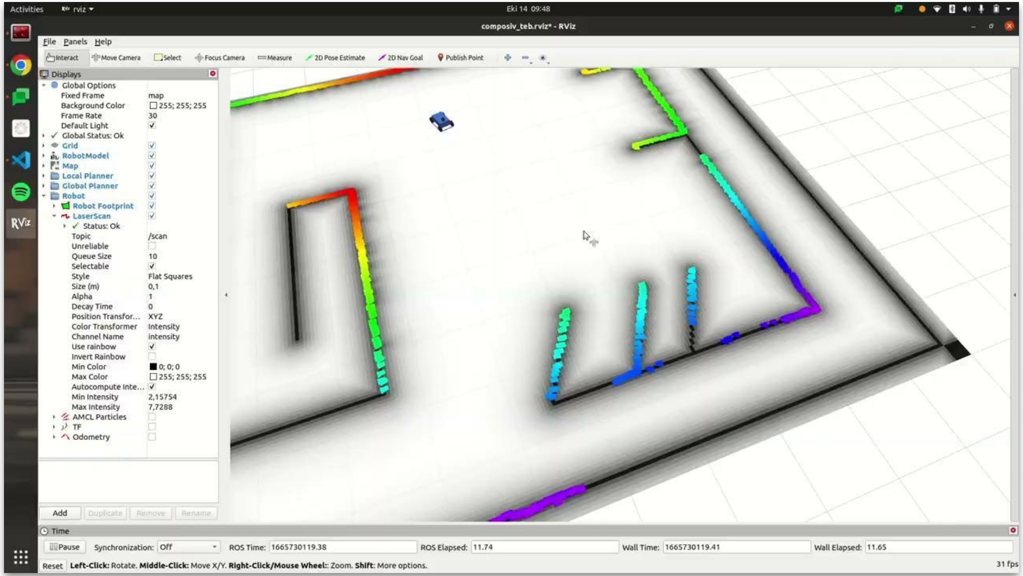
Stacks

Name includes

ai.composiv.sandbox.f1tenth:testcar_base.launch	Muto TestCar Base Stack	set apply
ai.composiv.sandbox.f1tenth:donkey_base.launch	Muto Donkeycar Base Stack	set apply
ai.composiv.sandbox.f1tenth:composiv_simulator_teb.launch	Muto Learning Simulator with TEB	set apply
ai.composiv.sandbox.f1tenth:composiv_simulator_gf.launch	Muto Learning Simulator with Gap Follwer	current start stop

Muto Dashboard

- Centralized management
  - Extensible and modular
  - µFrontends
  - Muto LiveUI
- Exemplars
  - ROS Inspection
  - Vehicles
  - Stack & Lifecycle Management
  - Telemetry



```
Global Options
  Fixed Frame              map
  Background Color         □ 255; 255; 255
  Frame Rate               30
  Default Light            ✓
✓ Global Status: Ok
Grid                       ✓
RobotModel                 ✓
Map                        ✓
Local Planner              ✓
Global Planner             ✓
Robot                      ✓
  Robot Footprint          ✓
  LaserScan                ✓
    ✓ Status: Ok
    Topic                  /scan
    Unreliable             □
    Queue Size             10
    Selectable             ✓
    Style                  Flat Squares
    Size (m)               0,1
    Alpha                  1
    Decay Time             0
    Position Transfor...   XYZ
    Color Transformer      Intensity
    Channel Name           intensity
    Use rainbow            ✓
    Invert Rainbow         □
    Min Color              ■ 0; 0; 0
    Max Color              □ 255; 255; 255
    Autocompute Inte...    ✓
    Min Intensity          2,15754
    Max Intensity          7,7288
  AMCL Particles           □
  TF                       □
  Odometry                 □
```

```
Synchronization: Off   ROS Time: 1665730119.38   ROS Elapsed: 11.74   Wall Time: 1665730119.41   Wall Elapsed: 11.65
```

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options.

[redacted]

muto liveUI

LiveUI helps you divide a monolithic frontend into smaller, more manageable micro frontends.

There is no magic, LiveUI allows you to split and manage your codebase, teams, release processes and runtimes independently

Composable at Runtime! LiveUI has the ability to change UI while its running

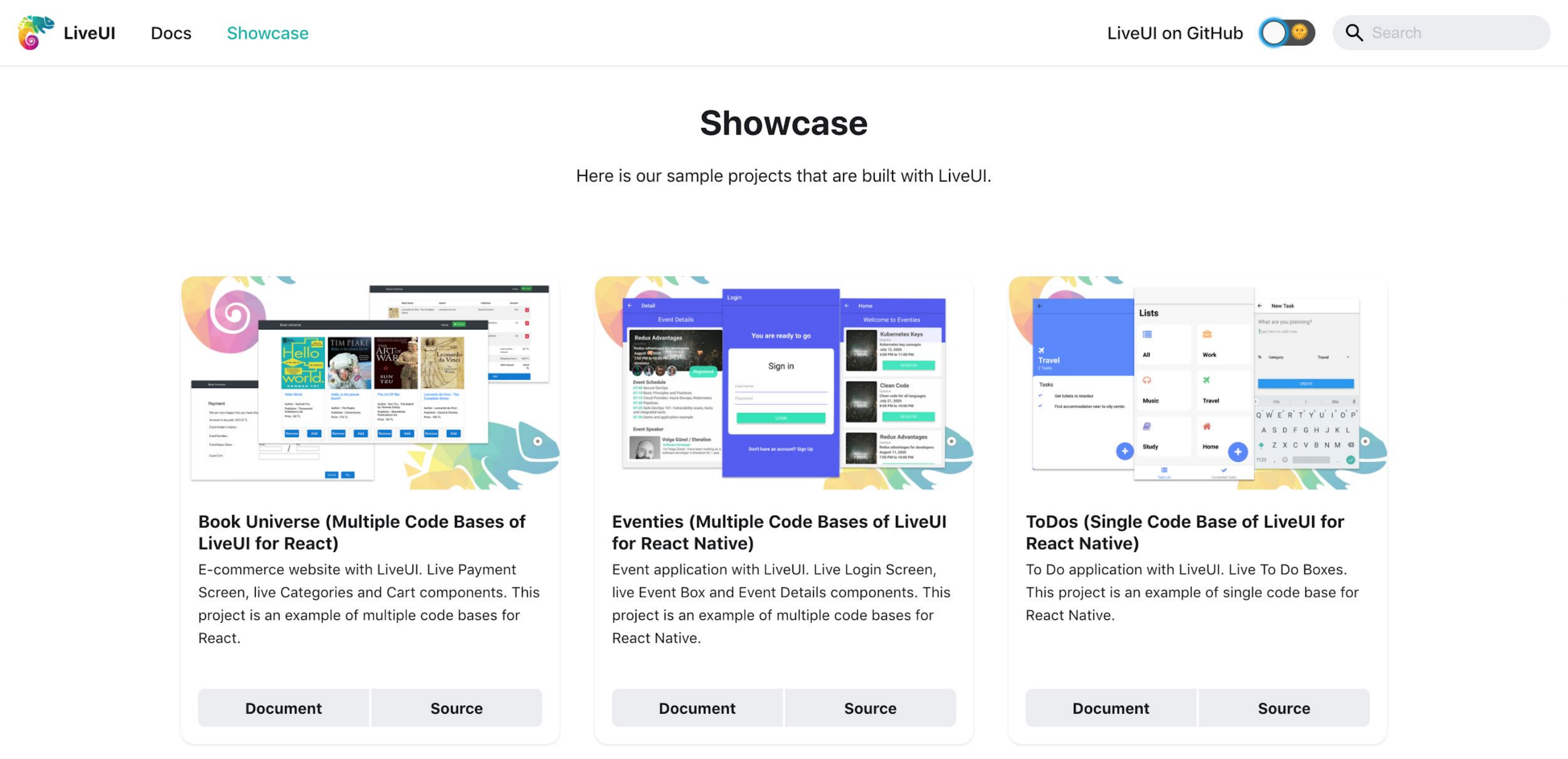
Framework Support

LiveUI is general purpose, framework and bundler agnostic.

LiveUI works with

- **React.js**
- **React Native**
- **Vue.js**
- **NativeScript**

You can use with the bundler of your choice such as **Webpack**, **Metro** and others. It will reuse support from the underlying stack such as the upcoming Module federation from webpack 5 automatically, or default to its own.



Showcase

Here is our sample projects that are built with LiveUI.

Book Universe (Multiple Code Bases of LiveUI for React)

E-commerce website with LiveUI. Live Payment Screen, live Categories and Cart components. This project is an example of multiple code bases for React.

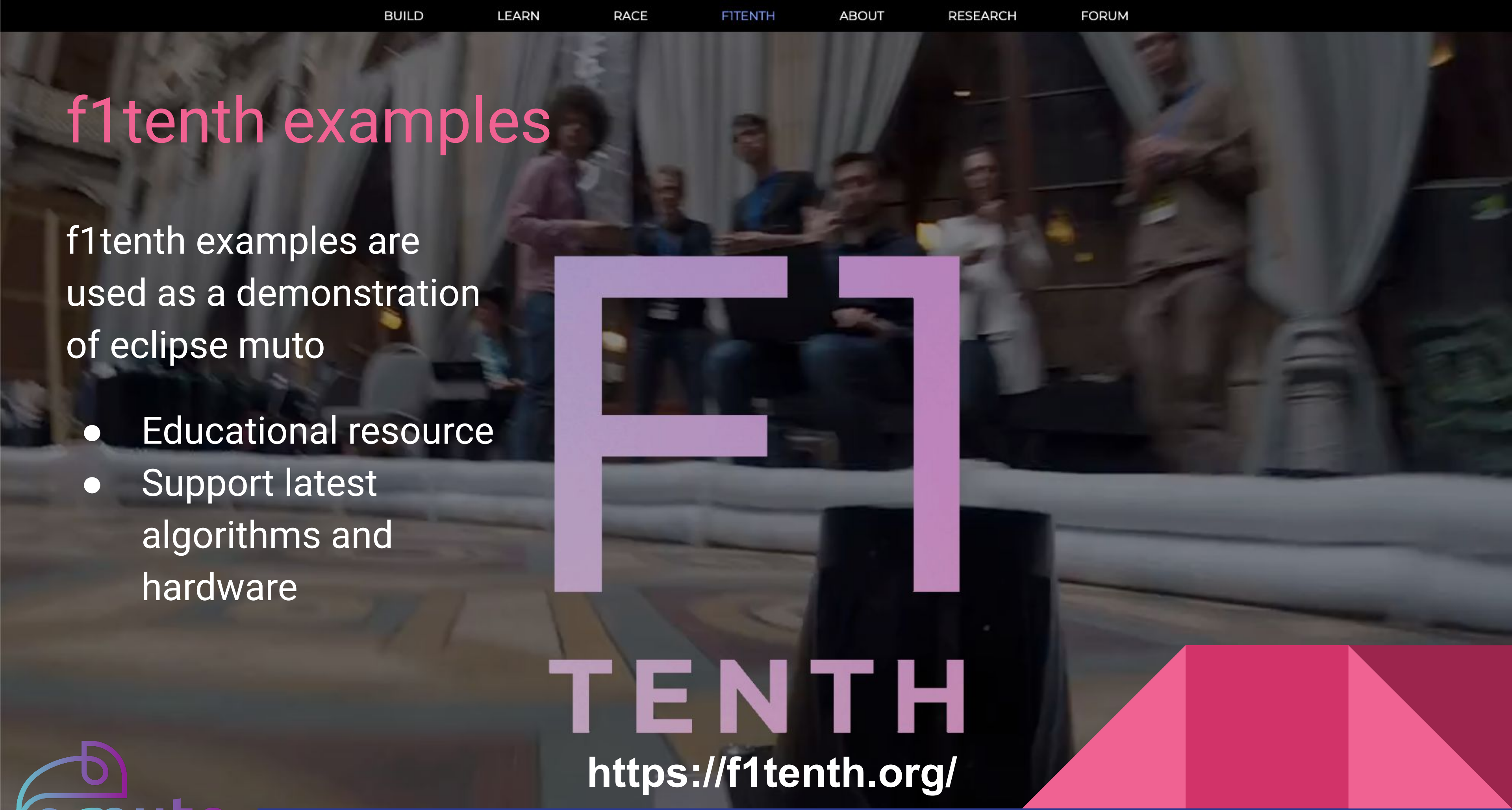
Eventies (Multiple Code Bases of LiveUI for React Native)

Event application with LiveUI. Live Login Screen, live Event Box and Event Details components. This project is an example of multiple code bases for React Native.

ToDos (Single Code Base of LiveUI for React Native)

To Do application with LiveUI. Live To Do Boxes. This project is an example of single code base for React Native.

f1tenth examples



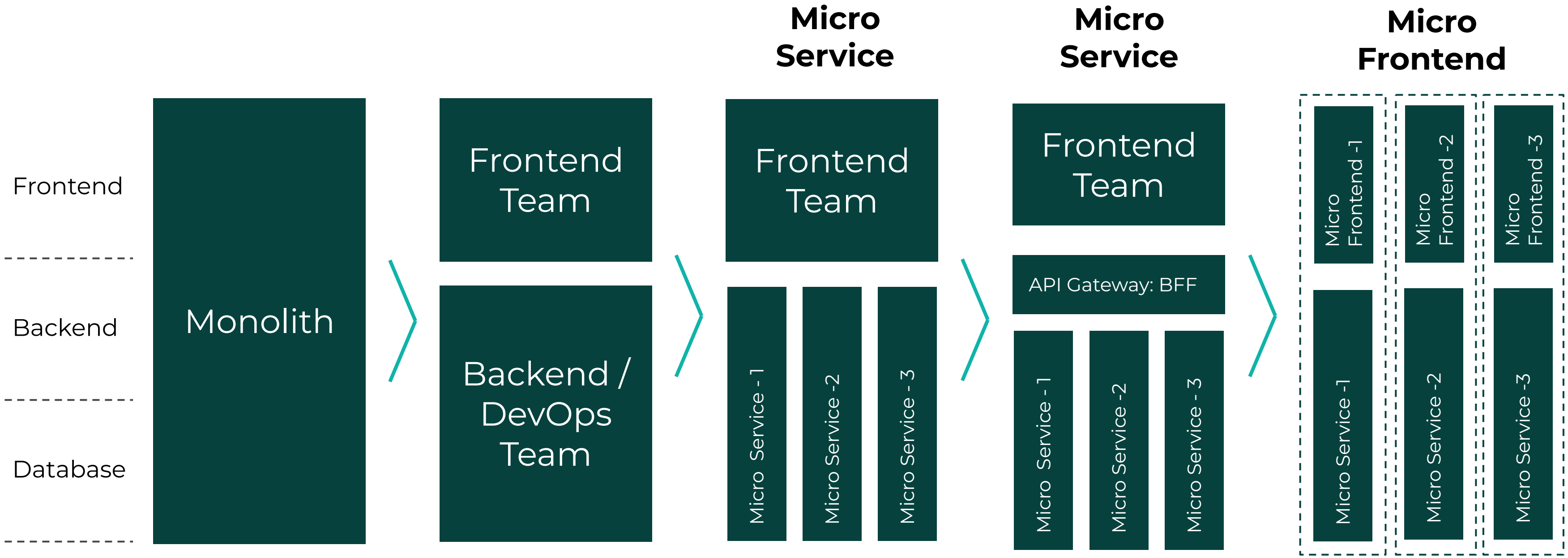
f1tenth examples are used as a demonstration of eclipse muto

- Educational resource
- Support latest algorithms and hardware

https://f1tenth.org/

muto

Micro Frontend ?



Independently Deployed

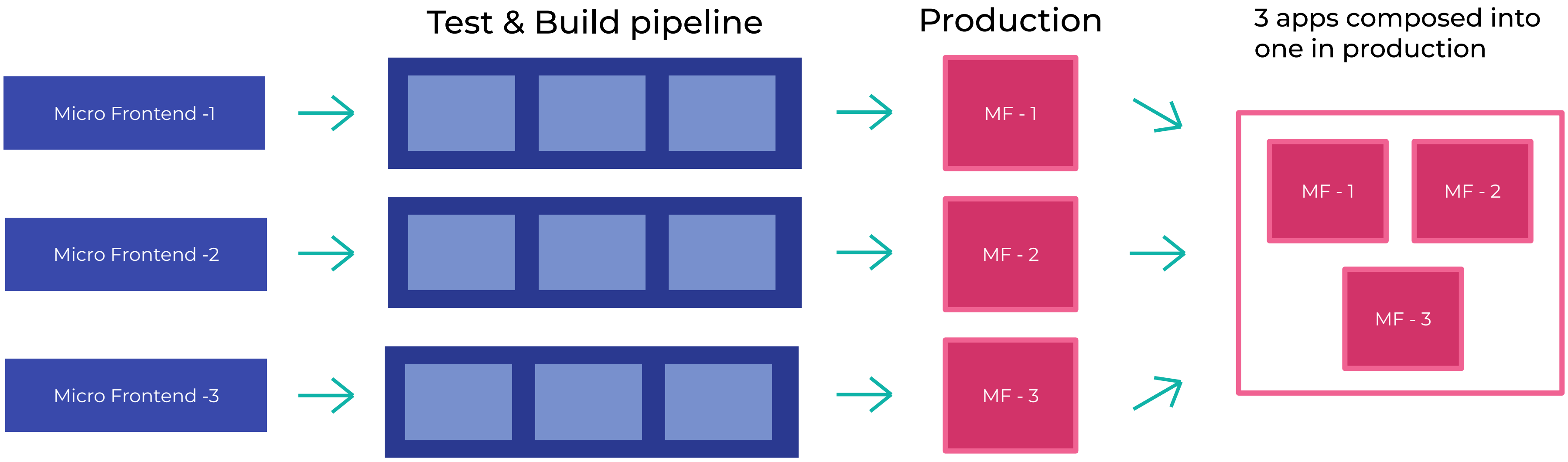
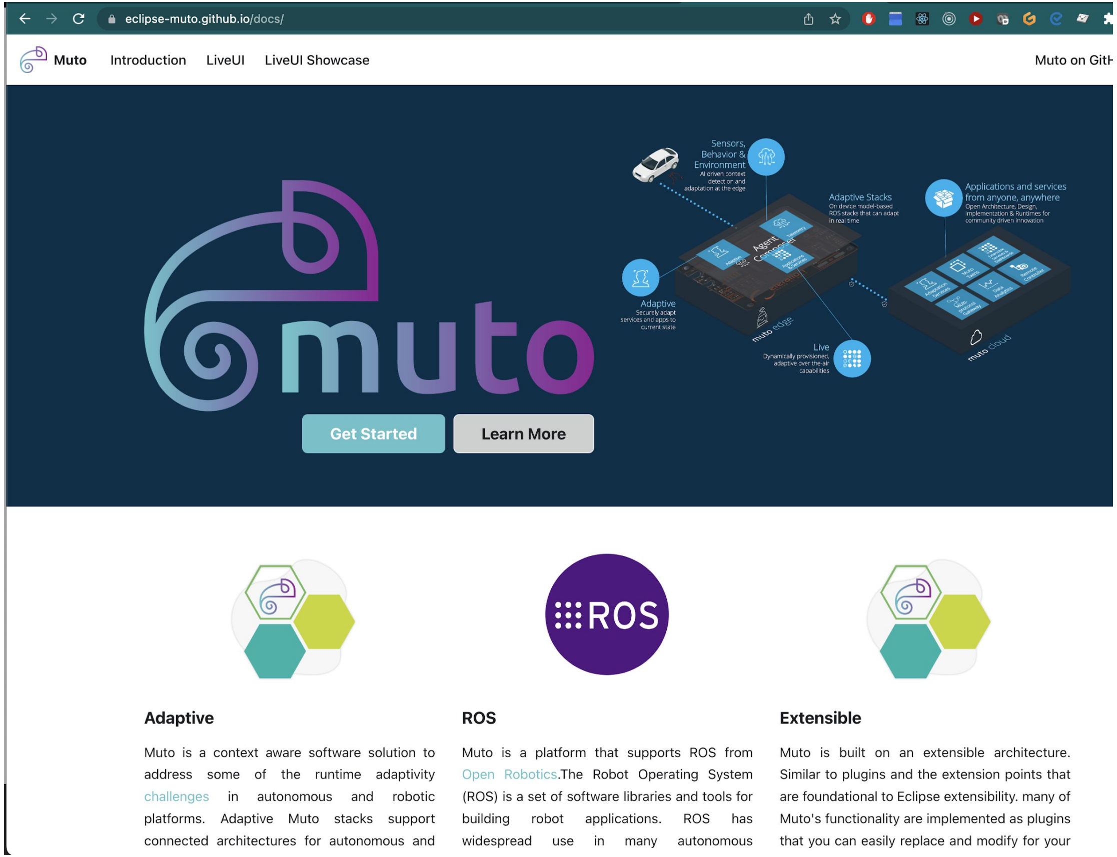


diagram reproduced from https://martinfowler.com/articles/micro-frontends.html

muto

https://eclipse-muto.github.io/docs/


eclipse-muto.github.io/docs/
Muto
Introduction
LiveUI
LiveUI Showcase
Muto on GitH
muto
Get Started
Learn More
Sensors, Behavior & Environment
AI driven context detection and adaptation at the edge
Adaptive Stacks
On device model-based ROS stacks that can adapt in real time
Applications and services from anyone, anywhere
Open Architecture, Design, Implementation & Runtimes for community driven innovation
Agent Composer
Telemetry
Adaptive
Applications & Services
Adaptive
Securely adapt services and apps to current state
muto edge
Live
Dynamically provisioned, adaptive over the-air capabilities
Extensible Services & Dashboards
Muto Twins
Remote Controller
Adaptation Services
Data Analytics
Multi-protocol Gateway
muto cloud
ROS

Adaptive

Muto is a context aware software solution to address some of the runtime adaptivity challenges in autonomous and robotic platforms. Adaptive Muto stacks support connected architectures for autonomous and

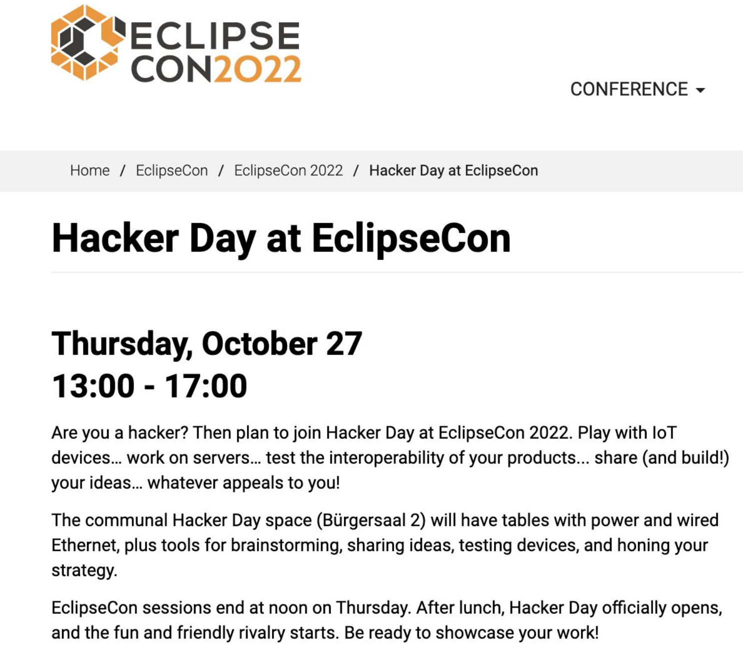
ROS

Muto is a platform that supports ROS from Open Robotics.The Robot Operating System (ROS) is a set of software libraries and tools for building robot applications. ROS has widespread use in many autonomous

Extensible

Muto is built on an extensible architecture. Similar to plugins and the extension points that are foundational to Eclipse extensibility. many of Muto's functionality are implemented as plugins that you can easily replace and modify for your

EclipseCON 2022


ECLIPSE CON2022
CONFERENCE
Home / EclipseCon / EclipseCon 2022 / Hacker Day at EclipseCon

Hacker Day at EclipseCon

Thursday, October 27
13:00 - 17:00

Are you a hacker? Then plan to join Hacker Day at EclipseCon 2022. Play with IoT devices… work on servers… test the interoperability of your products… share (and build!) your ideas… whatever appeals to you!

The communal Hacker Day space (Bürgersaal 2) will have tables with power and wired Ethernet, plus tools for brainstorming, sharing ideas, testing devices, and honing your strategy.

EclipseCon sessions end at noon on Thursday. After lunch, Hacker Day officially opens, and the fun and friendly rivalry starts. Be ready to showcase your work!

Join the Eclipse Software Defined Vehicle Hackathon Challenge at BCX22

November 7-9, 2022
Berlin

Meet members from the following Eclipse Automotive projects:

- Eclipse Chariott
- Eclipse eCAL
- Eclipe Kuksa
- Eclipse Leda
- Eclipse Muto
- Eclipse SommR
- Eclipse Velocitas

Scan for projects info

Register here:

muto

naci.dai@composiv.ai
deniz.memis@composiv.ai

Thank you !

- projects.eclipse.org/projects/automotive.muto
  - https://eclipse-muto.github.io/docs/
  - github.com/eclipse-muto

ECLIPSE
INCUBATION