Simplifying Cloud-Edge simulations with Python

ECLYPSE
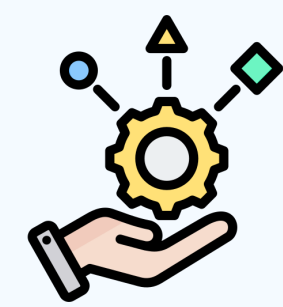
Valerio De Caro
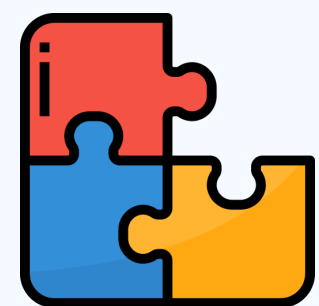valerio.decaro@di.unipi.it

Jacopo Massa
jacopo.massa@di.unipi.it

Department of Computer Science, University of Pisa, Italy

WHY ?

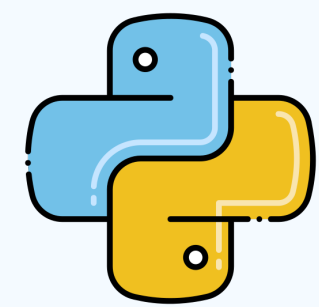- Cloud-Edge continuum is *heterogeneous*, *dynamic* and *QoS-sensitive*
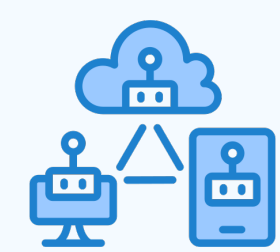
- Simulation is *not enough*

- Lack of *user-friendly* and *accessible* tools that easily capture the environment complexity
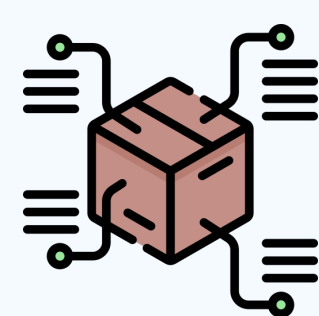
WHAT ?

- A Cloud-Edge *Python platform* for simulated/emulated runtime environments
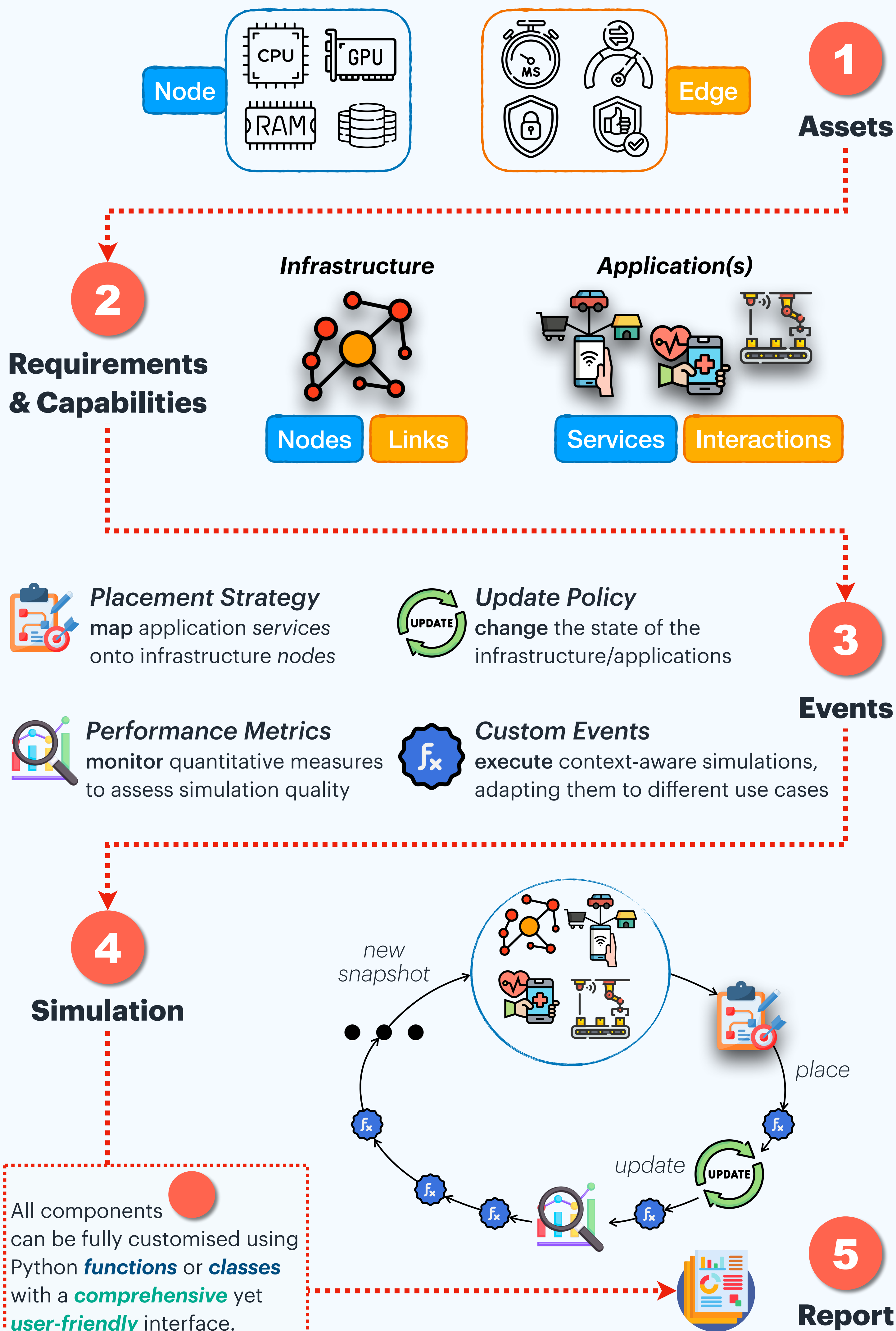
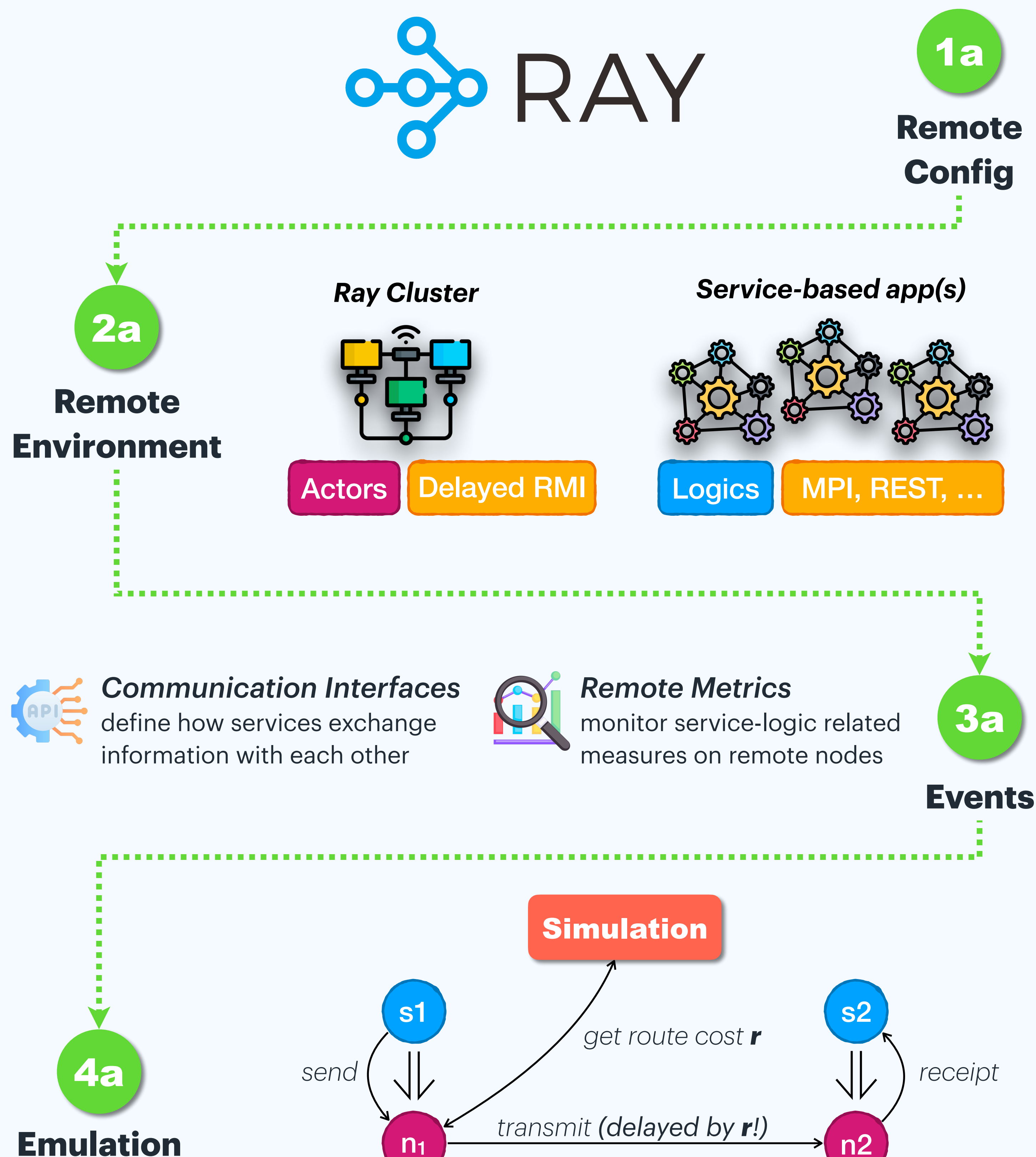- Designed to *manage*, *monitor* and *test* complex Cloud-Edge scenarios

- Allow the modelling of *real-world* infrastructures/application behaviours.

SIMULATION

1 Assets

Node: CPU, GPU, RAM — Edge: MS

2 Requirements & Capabilities

Infrastructure: Nodes | Links

Application(s): Services | Interactions

3 Events

Placement Strategy **map** application *services* onto infrastructure *nodes*

Update Policy **change** the state of the infrastructure/applications

Performance Metrics **monitor** quantitative measures to assess simulation quality

Custom Events **execute** context-aware simulations, adapting them to different use cases

4 Simulation

new snapshot — *place* — *update* (UPDATE)

All components can be fully customised using Python *functions* or *classes* with a *comprehensive* yet *user-friendly* interface.

5 Report

EMULATION

RAY

1a Remote Config

2a Remote Environment

Ray Cluster: Actors | Delayed RMI

Service-based app(s): Logics | MPI, REST, ...

3a Events

Communication Interfaces define how services exchange information with each other

Remote Metrics monitor service-logic related measures on remote nodes

4a Emulation

Simulation

s1 — *send* → n1 — *transmit (delayed by r!)* → n2

s1 ← *get route cost r* → s2

s2 — *receipt* → n2

KEEP US IN TOUCH

Email | GitHub | Paper

TRY ECLYPSE