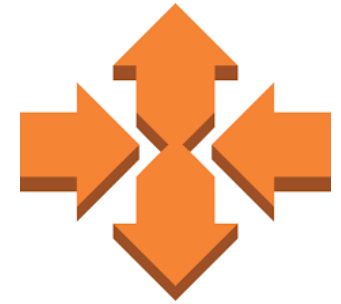
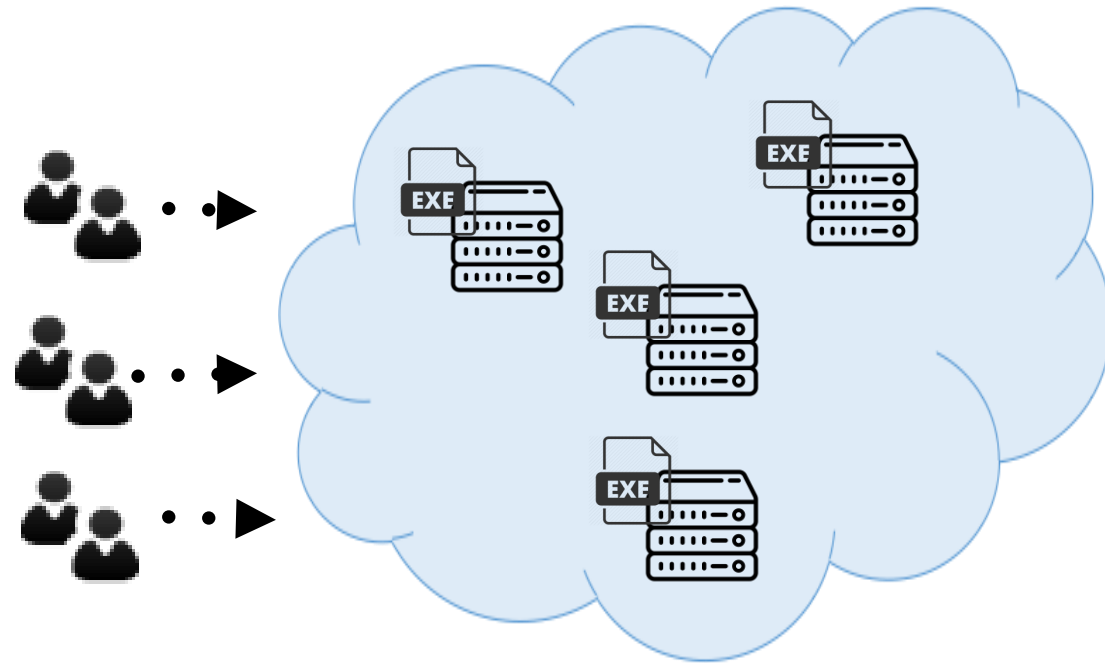


PLASMA: Programmable Elasticity for Stateful Cloud Computing Applications

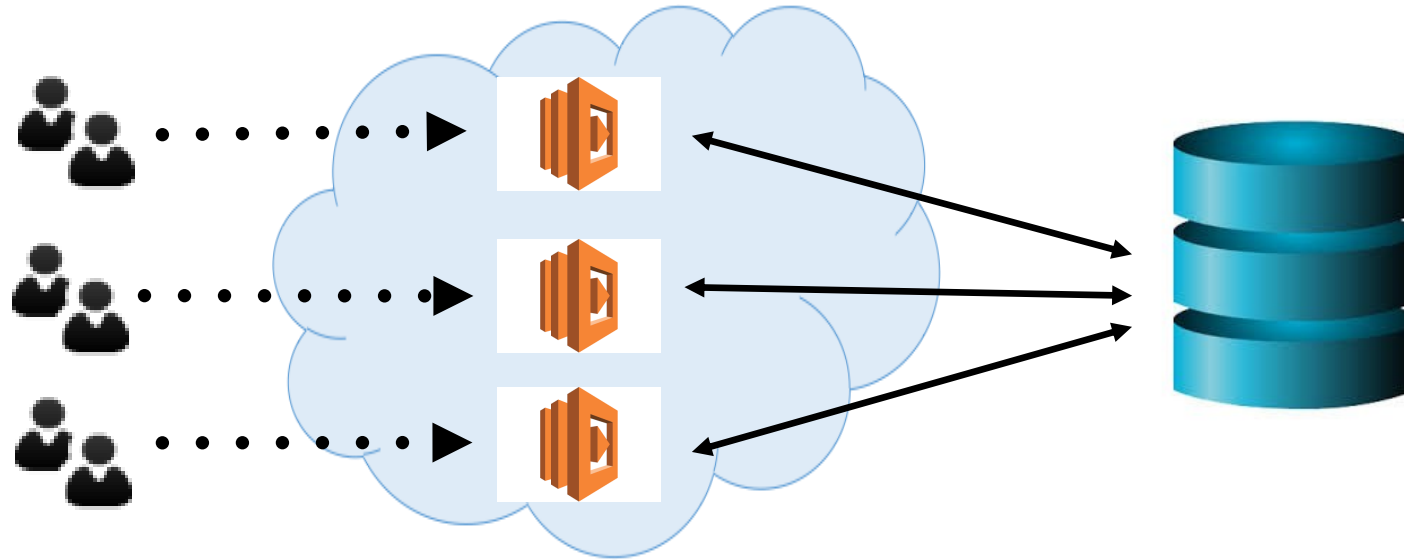
Bo Sang (Purdue University, Ant Financial Services Group),
Pierre-Louis Roman, Patrick Eugster (Università della Svizzera italiana),
Hui Lu (Binghamton University),
Srivatsan Ravi (University of Southern California),
Gustavo Petri (ARM Research)



Elasticity Management for Cloud Applications



AWS Lambda Function

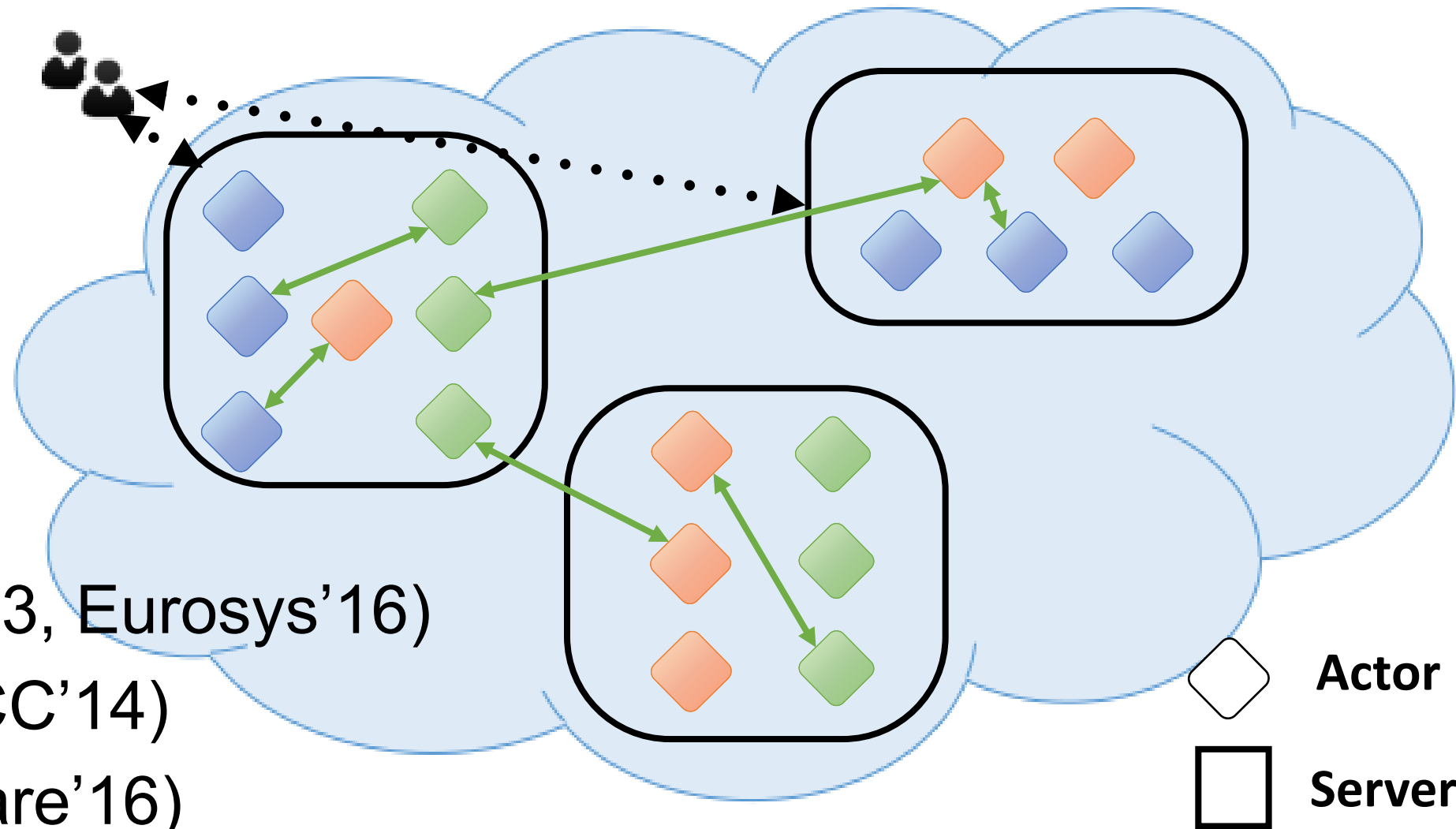


Each function executes independently

External storage introduce nontrivial latency

Actor-based Applications in Cloud

Scalability ✓
Low latency ✓
Elasticity ?



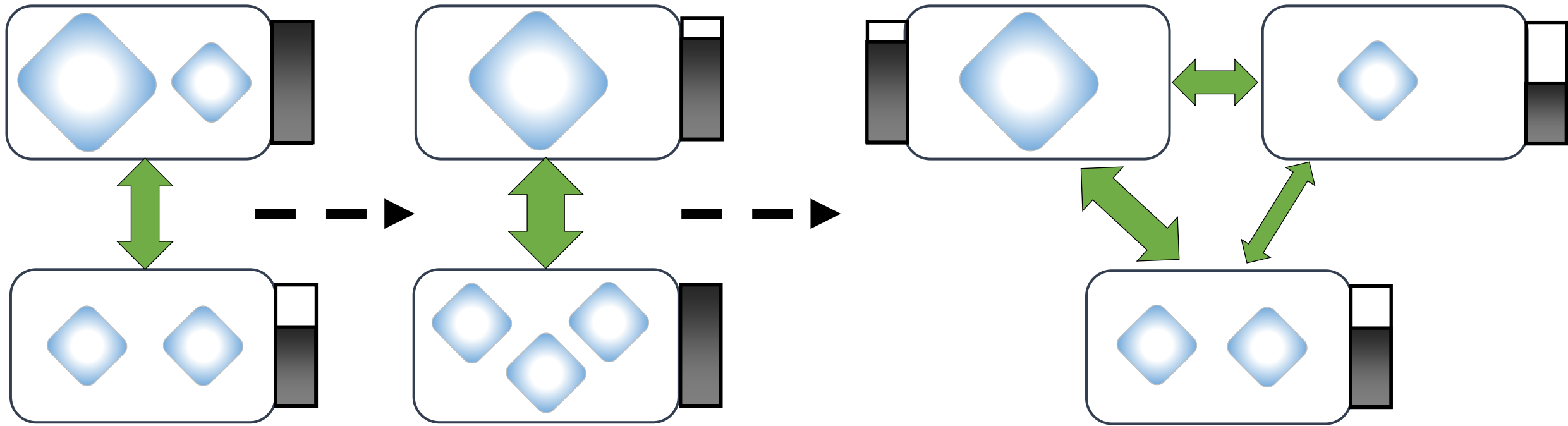
Orleans (SoCC'13, Eurosys'16)

EventWave (SoCC'14)

AEON (Middleware'16)

Actor
Server

Elasticity Management for PageRank



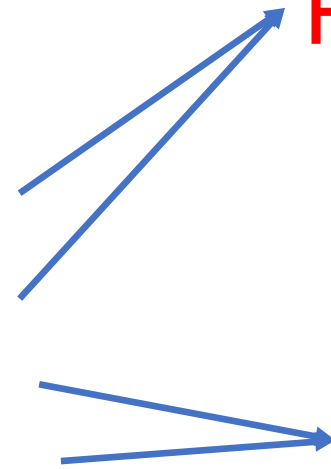
↔ Communication ◆ Graph partition □ Server 🔋 Resource Usage

Fine-grained Elasticity Management

- ❑ We need
 - ❑ Application information
 - ❑ User requirements
 - ❑ Server runtime information
 - ❑ Application runtime information

PLASMA Language

PLASMA Runtime



```

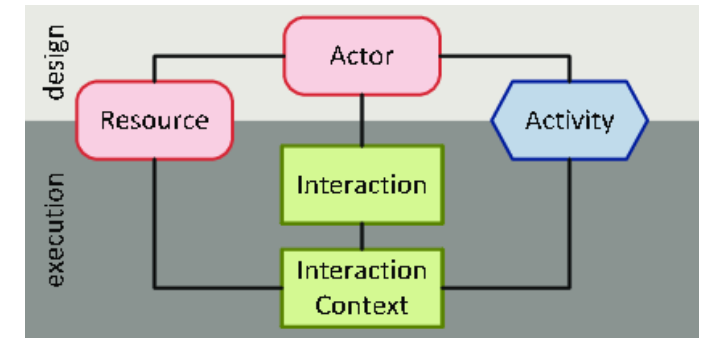
252 long fdoSteps(long target, int mdelay_2) { // The Movement Engine
253     long steps = 0;
254     char key = keypad.getKey();
255
256     steps = fsof(steps);
257     while (target >= steps) {
258         digitalWrite(pinClockplus, 1);
259         // delayMicroseconds(mdelay_1);
260         digitalWrite(pinClockplus, 0);
261         delayMicroseconds(mdelay_2);
262         steps++;
263     }
264     steps = fsoftop(target, steps);
265     // if (digitalRead(STOP) == 1) {
266         // fstop();
267         // target = 0;
268         // fscreen(0, "HIT STOP");
269         // delay(1000);
270     }
271 }
272 return steps;
273 // Set the amount to be moved in mm
274 long fset_target() {
275     long tempTarget = 0;
276     char key = '0';
277     String varskey = "";
278     fscreen(0, "Distance in mm");
279
280
281

```

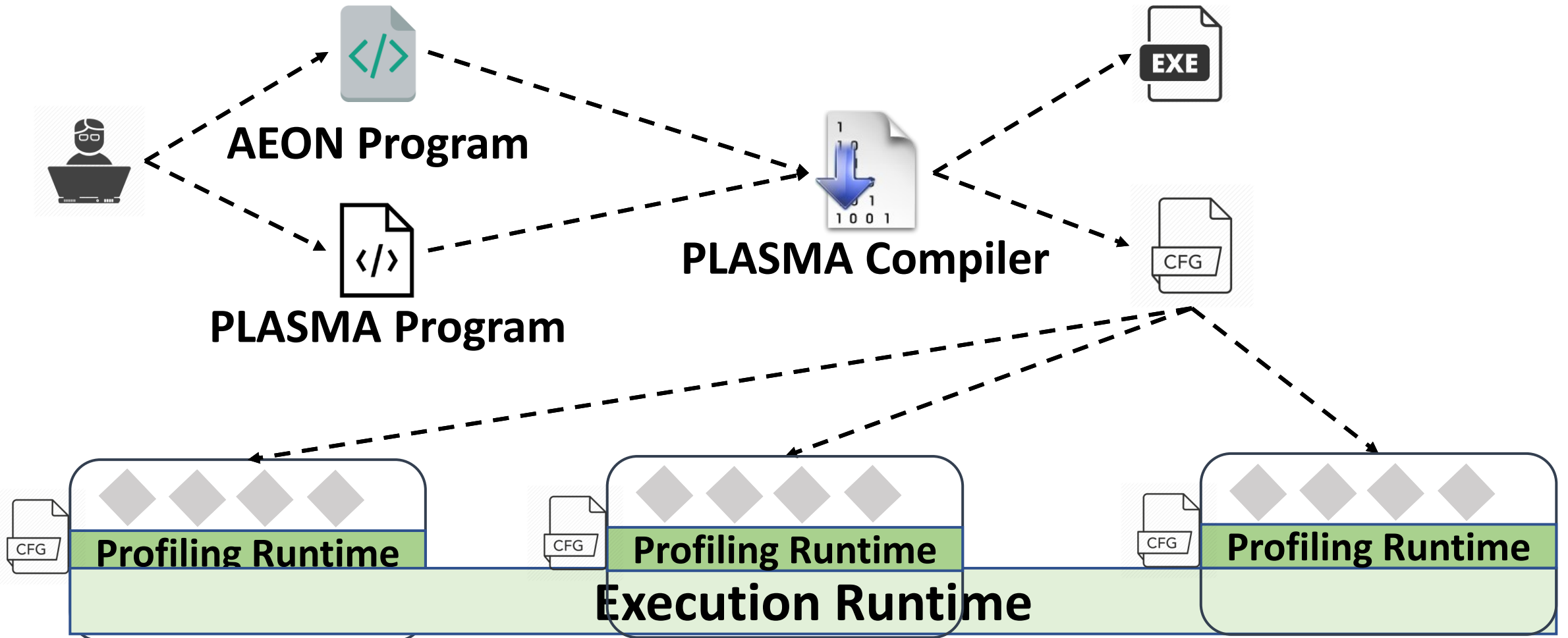
Requirements



Parameter	Specification
SoC	Broadcom BCM2837
CPU	4 ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
Networking	10/100 Ethernet 2.4GHz 802.11n
Bluetooth	4.1 Classic
Bluetooth	Low Energy
Storage	microSD



PLASMA Tool Chain



Elasticity Programming Language

□ Elasticity rules

Conds => Behaviors;

```
server.cpu.perc > 80 or server.cpu.perc < 60 =>  
balance({Partition}, cpu);
```

□ Conditions

Server runtime

Actor runtime

Semantics

...

balance(*{atypes}*, *resource*)

reserve(*actor*, *resource*)

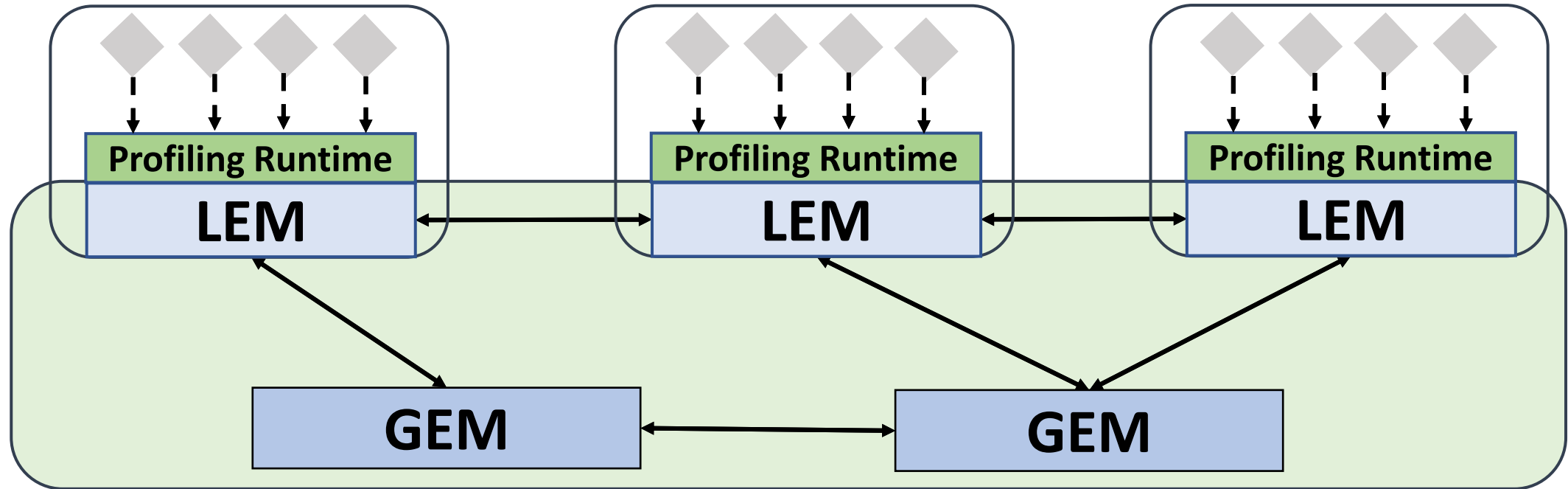
colocate(*actor*, *actor*)

separate(*actor*, *actor*)

pin(*actor*)

□ Behaviors:

Elasticity Management Runtime



 Profiling Runtime  Elasticity Messages  Execution Runtime

Profiling Runtime: collecting runtime information of actors and the server

LEM: processing rules which only require local information

GEM: processing rules which only require global information

Evaluation: Applications

Applications	Elasticity rules
Metadata server	<ol style="list-style-type: none">1. Colocate Folder with Files on the same server
PageRank	<ol style="list-style-type: none">1. Balance CPU workload
E-Store	<ol style="list-style-type: none">1. Put hot Partitions on idle servers2. Colocate parent-child Partitions3. Balance CPU workload of Partitions
Media Service	<ol style="list-style-type: none">1. Balance network workload for FrontEndsService2. Provide VideoStream with enough CPU3. Colocate linked VideoStream and UserInfo4. Avoid migrating MovieReview5. Balance CPU workload of ReviewChecker6. Colocate linked ReviewEditor and UserReview
Halo Presence Server	<ol style="list-style-type: none">1. Balance CPU workload of Routers2. Colocate Session with Players in it

Evaluation: PageRank

□ Setup

- SNAP's LiveJournal social network
- Use METIS to partition the graph into 32 partitions

24% faster with 16 vCPU (8 servers) 24 vCPU (12 servers) vs 32 vCPU (16 servers)

