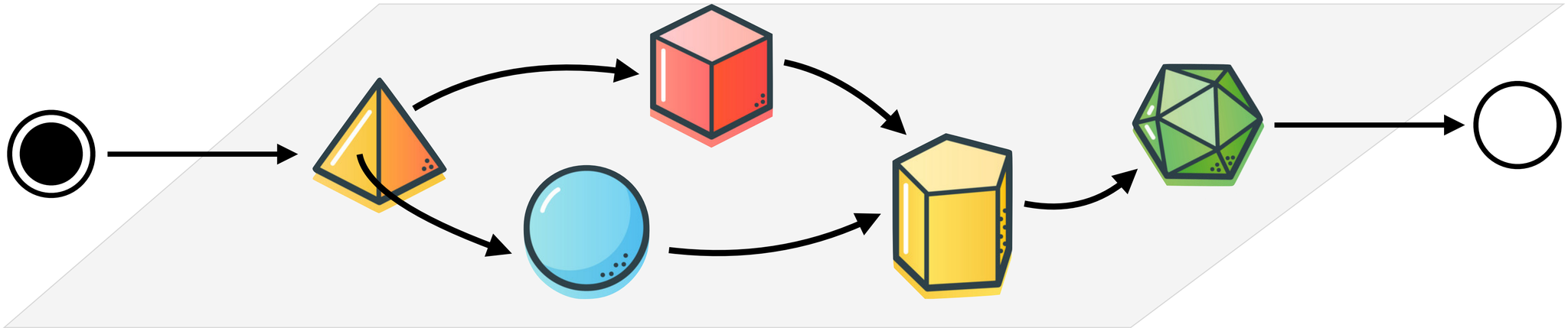
Serverless Computing in OLAS: Declarative Scheduling and Edge Cloud Platforms

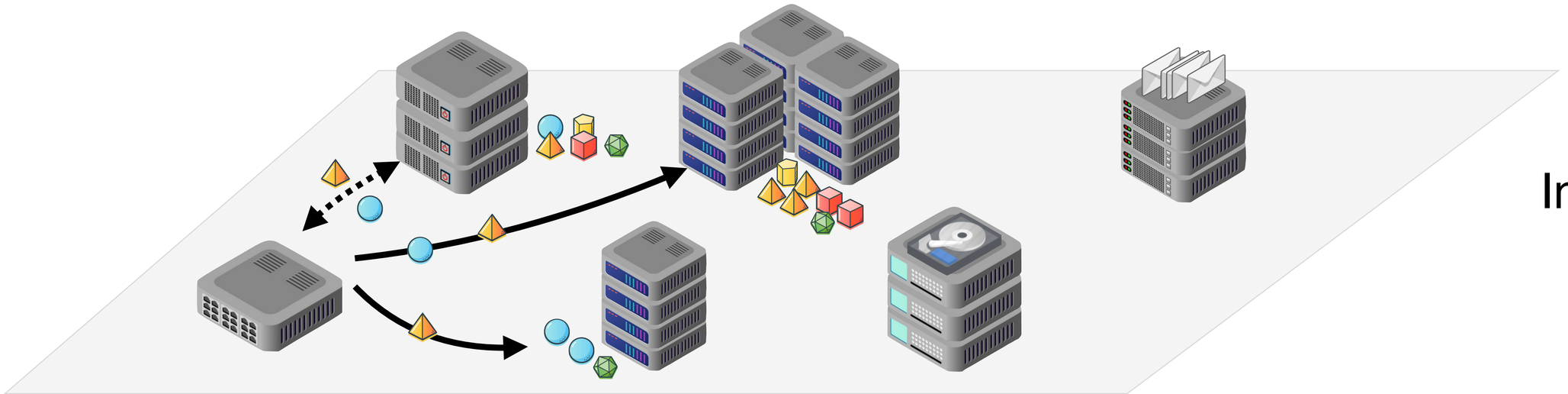
Saverio Giallorenzo

Università di Bologna (IT) and OLAS Team, INRIA (FR)

A Gentle Introduction to Serverless

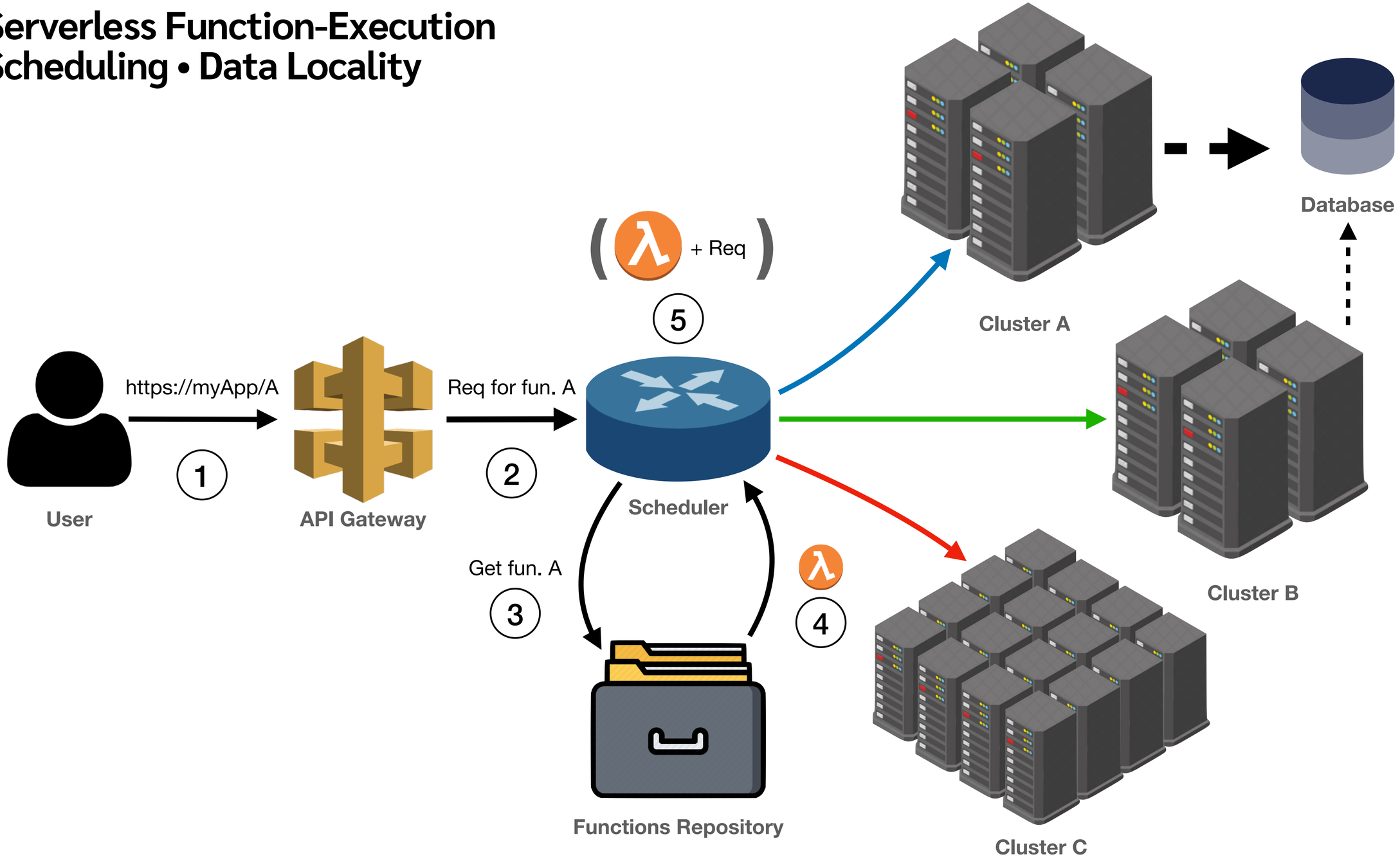


Programming Layer

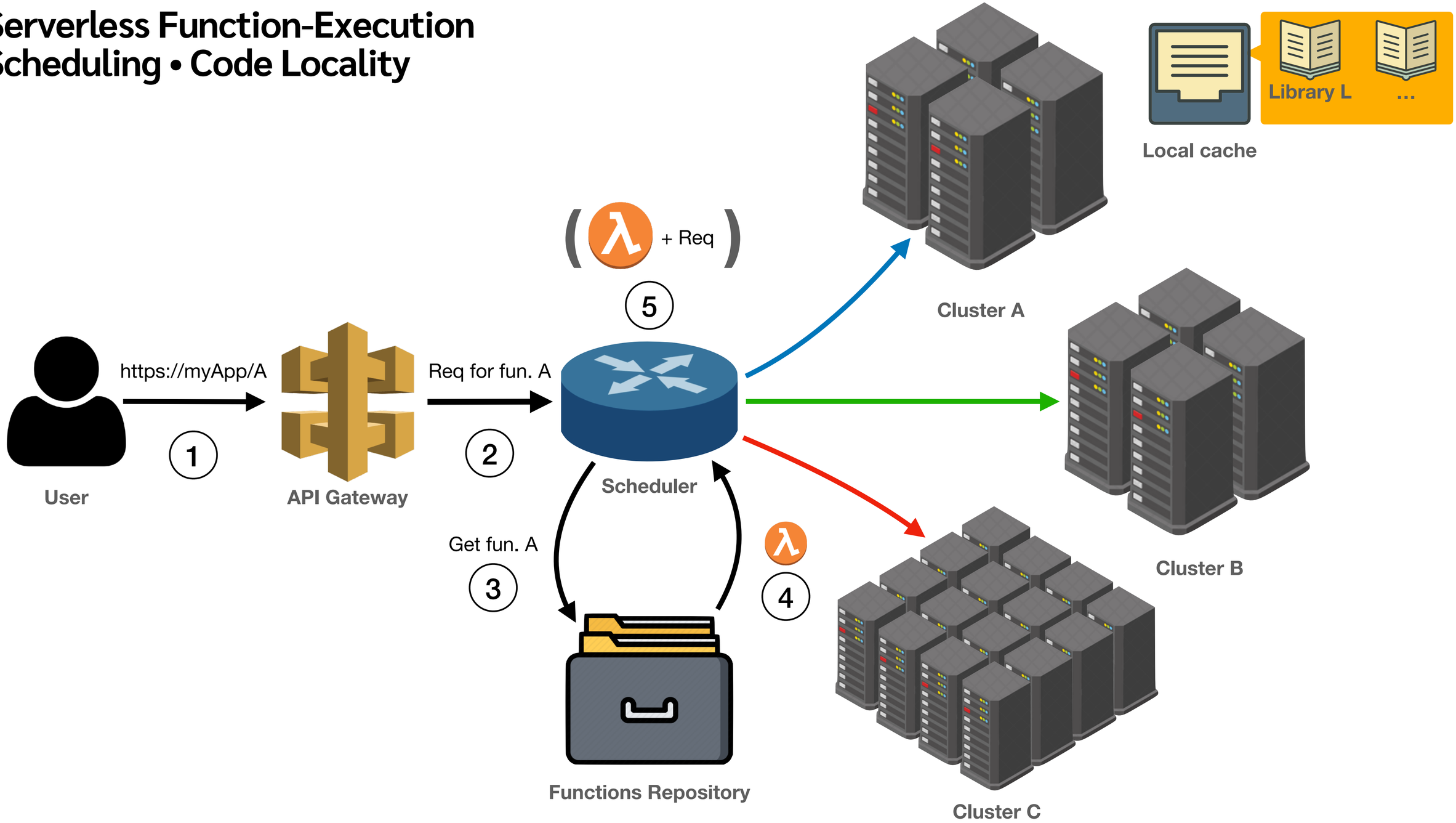


Implementation Layer

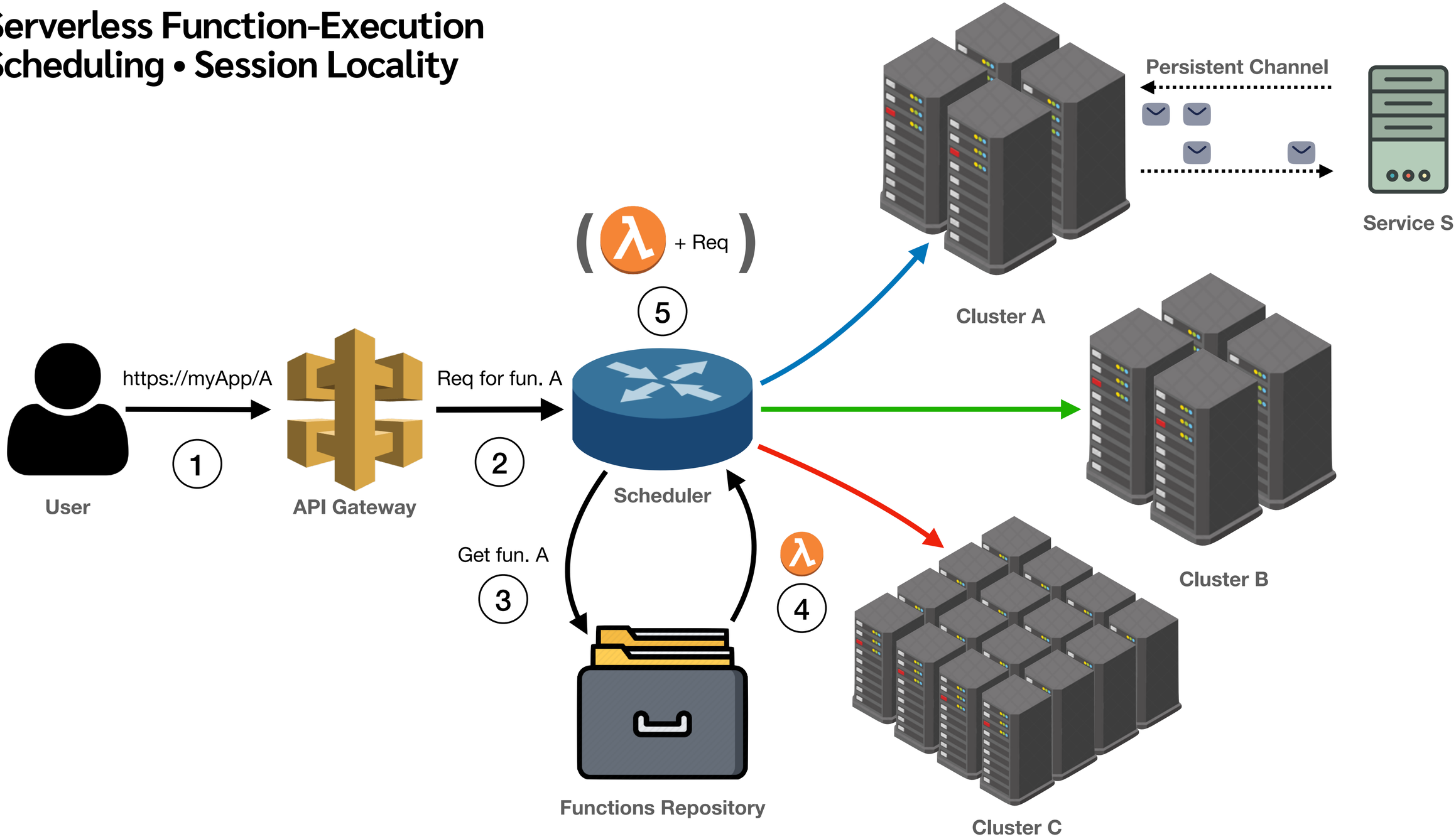
Serverless Function-Execution Scheduling • Data Locality



Serverless Function-Execution Scheduling • Code Locality

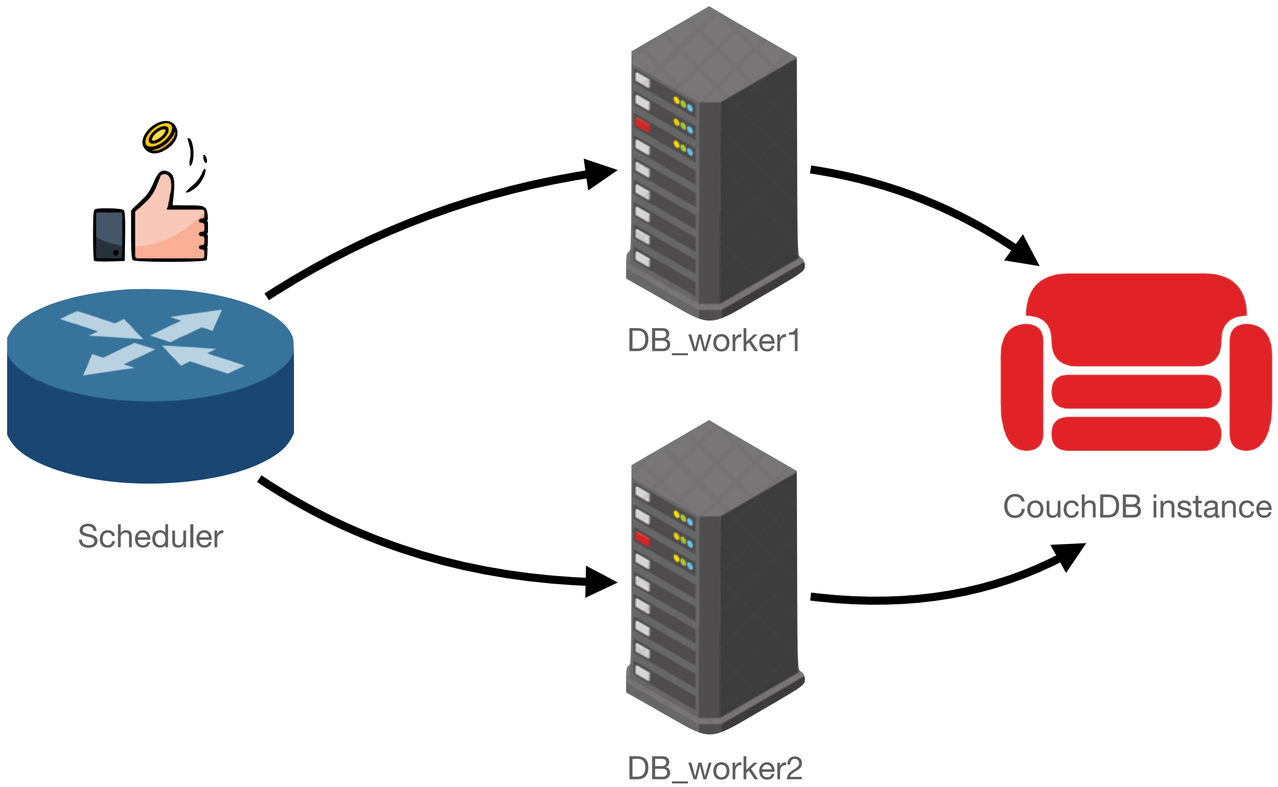


Serverless Function-Execution Scheduling • Session Locality



The APP Language • an example

```
couchdb_query:
  - workers:
      - DB_worker1
      - DB_worker2
    strategy: random
    invalidate: ↩
      capacity_used: 50%
    followup: fail
```



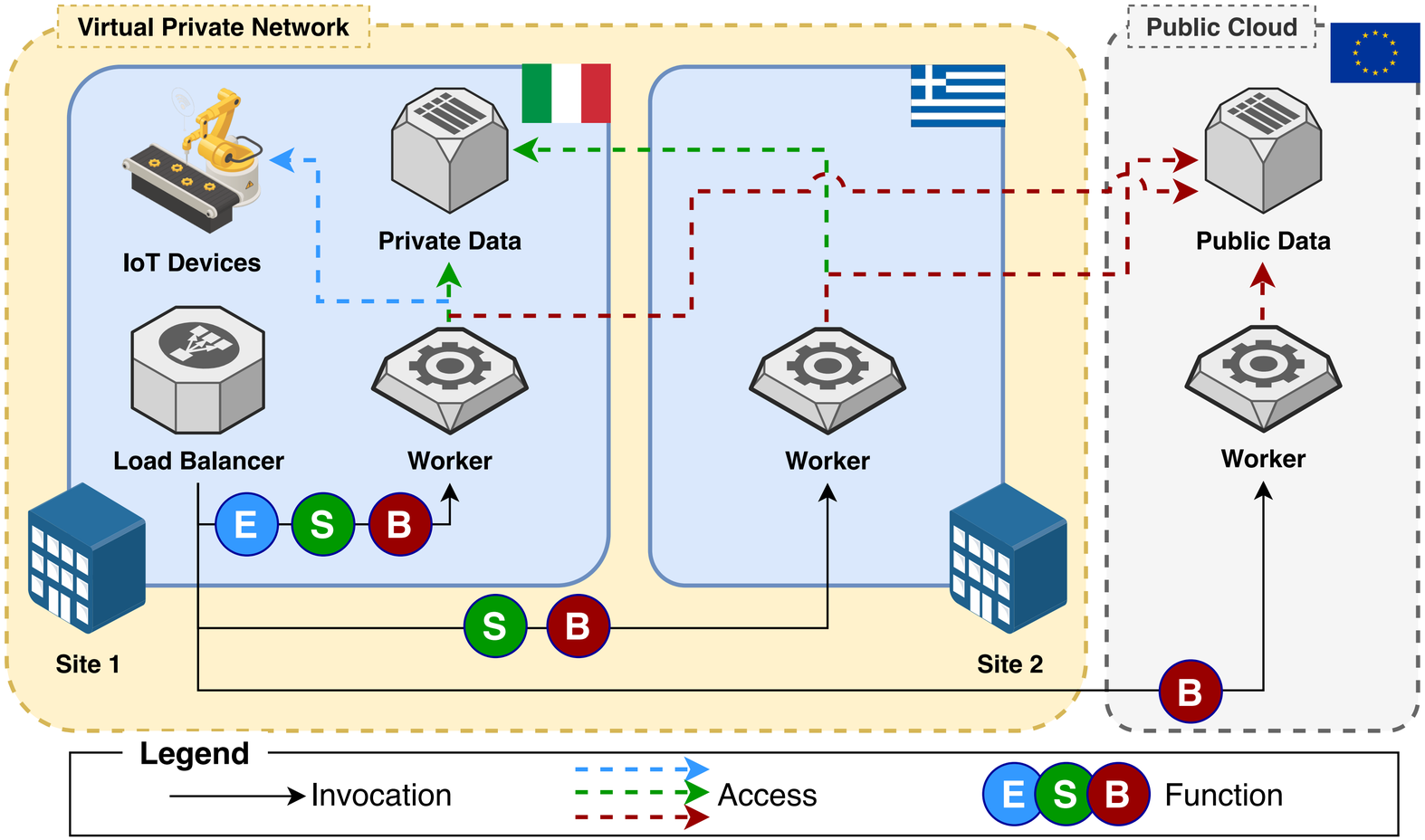
De Palma, G., Giallorenzo, S., Mauro, J., & Zavattaro, G. (2020). **Allocation Priority Policies for Serverless Function-Execution Scheduling Optimisation**. In E. Kafeza, B. Benatallah, F. Martinelli, H. Hacid, A. Bouguettaya, & H. Motahari (Eds.), Service-Oriented Computing - 18th International Conference, **ICSOC 2020**, Dubai, United Arab Emirates, December 14-17, 2020, Proceedings (pp. 416–430). Springer.

The APP Language • Case Study

```
Function_E:
  - workers:
    - worker_site1
  followup: fail

Function_S:
  - workers:
    - worker_site2
    - worker_site1
    strategy: random
  followup: fail

Function_B:
  - workers:
    - worker_public_cloud
    - worker_site2
    - worker_site1
  strategy: best_first
  followup: fail
```



The APP Language • Case Study, results

	Site 1	Site 2	Public Cloud	Average (ms)	95% Average (ms)
E	1000	0	0	1096.53	1019.03
S	466	534	0	149.18	90.86
B	0	90	910	105.18	64.62

Table 1. 1000 invocation for each function in the APP-based OpenWhisk deployment.

		Site 1	Site 2	Public Cloud	Average (ms)	95% Average (ms)
OW1	E	1000	0	0	1159.90	1025.52
OW2	S	19	981	0	385.30	302.08
OW3	B	185	815	0	265.69	215.793

Table 2. 1000 invocations for each function in the vanilla OpenWhisk deployment.

Cost-aware APP

```
1 ( isPremiumUser, par ) => {
2   if( isPremiumUser ) {
3     call PremiumService( par )
4   } else {
5     call BasicService( par )
6   }
7 }
```

MiniSL + cAPP

$$
\begin{aligned}
main(u,v,P,B) &= if_2(u,P,B) && [\,] \\
if_2(u,P,B) &= P && [\,u=1\,] \\
if_2(u,P,B) &= B && [\,u=0\,]
\end{aligned}
$$

```
- premUser:
  - workers:
      - wrk: W1
      - wrk: W2
    strategy: min_latency
```

De Palma, G., Giallorenzo, S., Laneve, C., Mauro, J., Trentin, M., & Zavattaro, G. (2024). **Leveraging static analysis for cost-aware serverless scheduling policies**. Int. J. Softw. Tools Technol. Transf., 26(6), 781–796.

Cost-aware APP

```
// name: lambda1.miniSL
// tag: premUser
( isPremiumUser, par ) => {
  if( isPremiumUser ) {
    call PremiumService( par )
  } else {
    call BasicService( par )
  }
}
```

```
// name: lambda2.miniSL
// tag: premUser
( username, par ) => {
  if( call IsPremiumUser(username)){
    call PremiumService( par )
  } else {
    call BasicService( par )
  }
}
```

cAPP script

```
- premUser:
  - workers:
      - wrk: W1
      - wrk: W2
    strategy: min_latency
```

Cost-aware APP

DEPLOYMENT TIME

```
// name: lambda1.miniSL
// tag: premUser
( isPremiumUser, par ) => {
  if( isPremiumUser ) {
    call PremiumService( par )
  } else {
    call BasicService( par )
  }
}
```

```
// name: lambda2.miniSL
// tag: premUser
( username, par ) => {
  if( call IsPremiumUser(username)){
    call PremiumService( par )
  } else {
    call BasicService( par )
  }
}
```

Inference of Cost Programs

$$main(u,v,P,B) = if_2(u,P,B) \quad [\,]$$
$$if_2(u,P,B) = P \quad [\,u=1\,]$$
$$if_2(u,P,B) = B \quad [\,u=0\,]$$

$$main(u,v,K,P,B) = K + max(P,B) \quad [\,]$$

cAPP script

```
- premUser:
  - workers:
      - wrk: W1
      - wrk: W2
    strategy: min_latency
```

Cost-aware APP

```
// name: lambda1.miniSL
// tag: premUser
( isPremiumUser, par ) => {
  if( isPremiumUser ) {
    call PremiumService( par )
  } else {
    call BasicService( par )
  }
}
```

```
// name: lambda2.miniSL
// tag: premUser
( username, par ) => {
  if( call IsPremiumUser(username)){
    call PremiumService( par )
  } else {
    call BasicService( par )
  }
}
```

Also, `max_latency: n` for invalidate

cAPP script

```
- premUser:
  - workers:
      - wrk: W1
      - wrk: W2
    strategy: min_latency
```

Inference of Cost Programs

$$main(u,v,P,B) = if_2(u,P,B) \quad [\,]$$
$$if_2(u,P,B) = P \quad [\,u=1\,]$$
$$if_2(u,P,B) = B \quad [\,u=0\,]$$

$$main(u,v,K,P,B) = K + max(P,B) \quad [\,]$$

DEPLOYMENT TIME

SCHEDULING TIME

Request for lambda1(1,v)

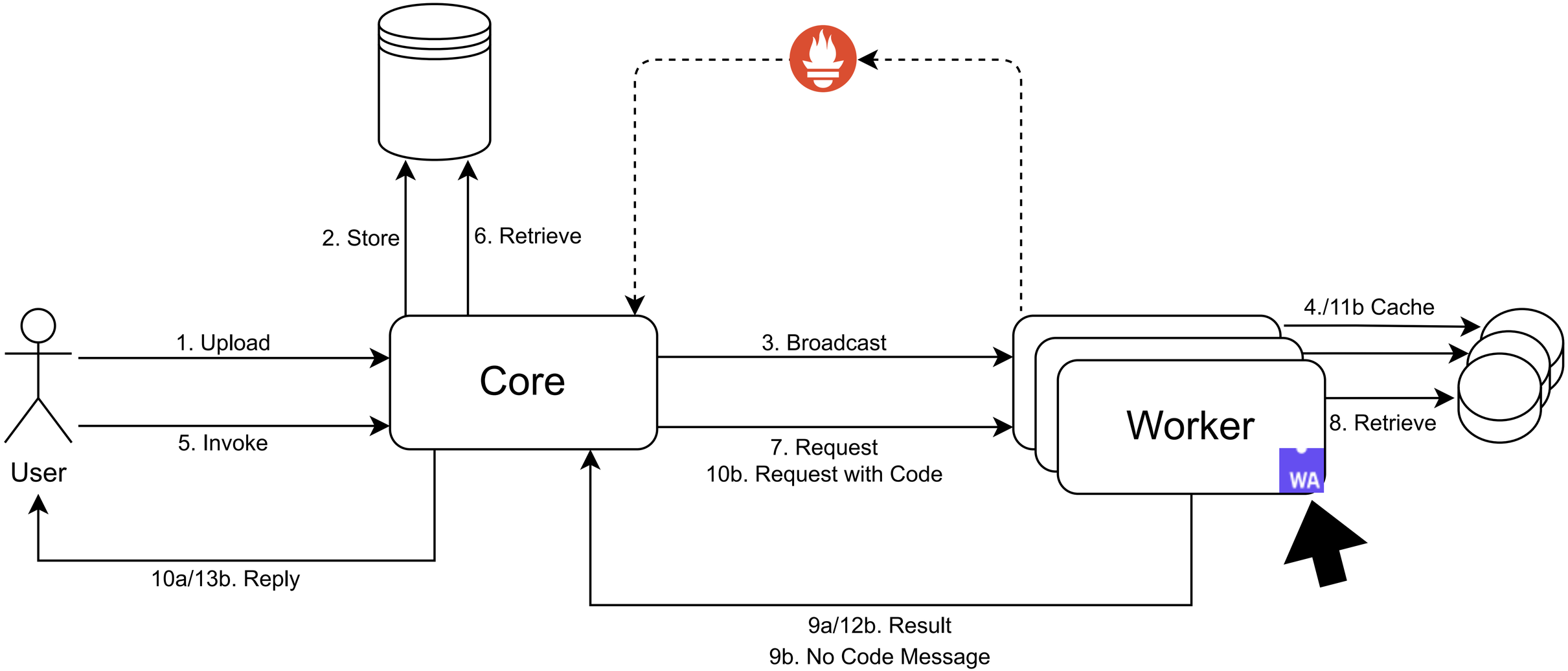
```
W in ( W1, W2 )
 where W.latency( PremiumService )
 is minimal
```

Request for lambda2(u_name,v)

```
W in ( W1, W2 )
 where W.latency( IsPremiumUser )
  + max( W.latency( PremiumService ),
         W.latency( BasicService ) )
 is minimal
```

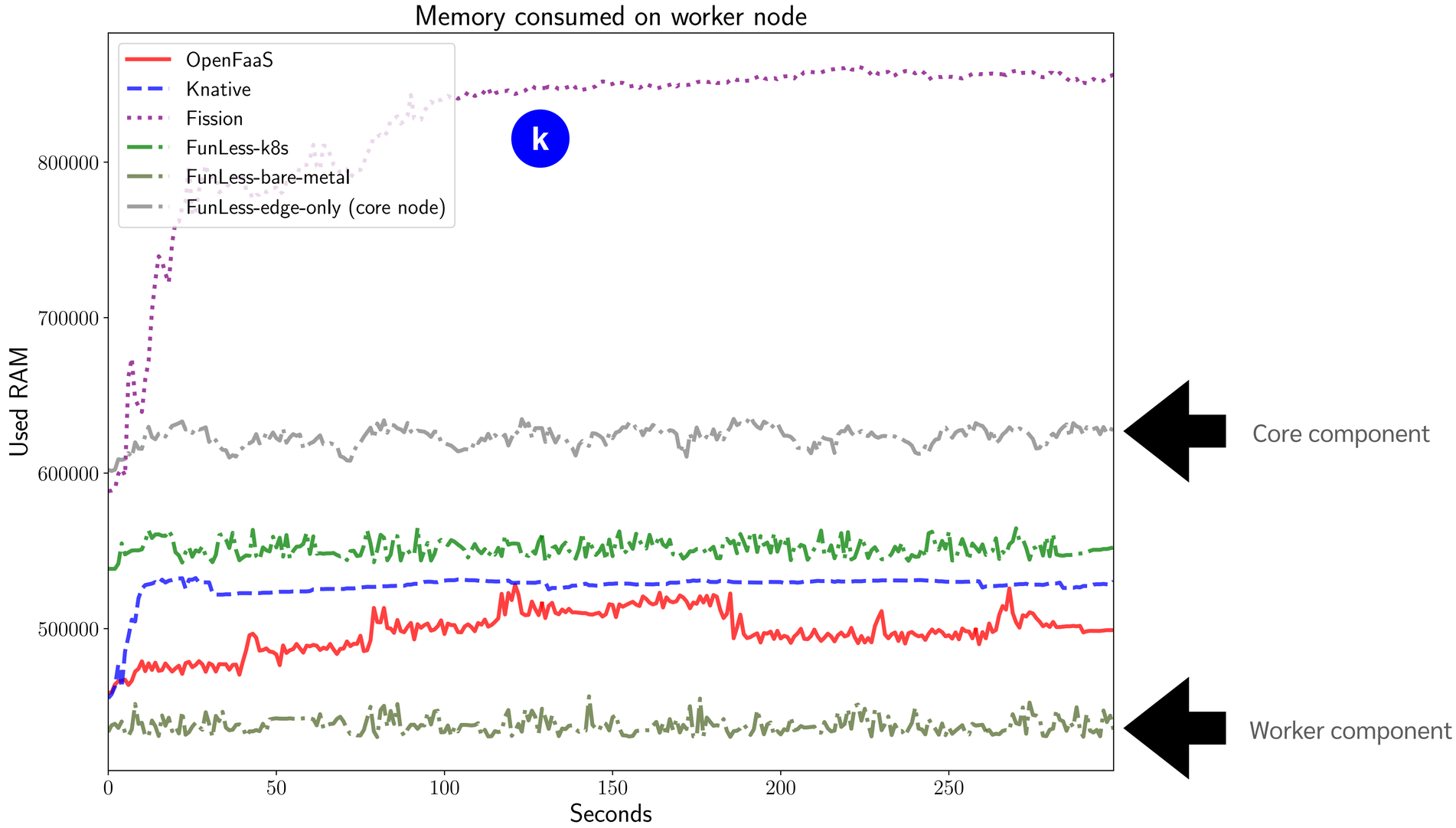
Cost Program Solver

FunLess: Extending Serverless to Edge Computing



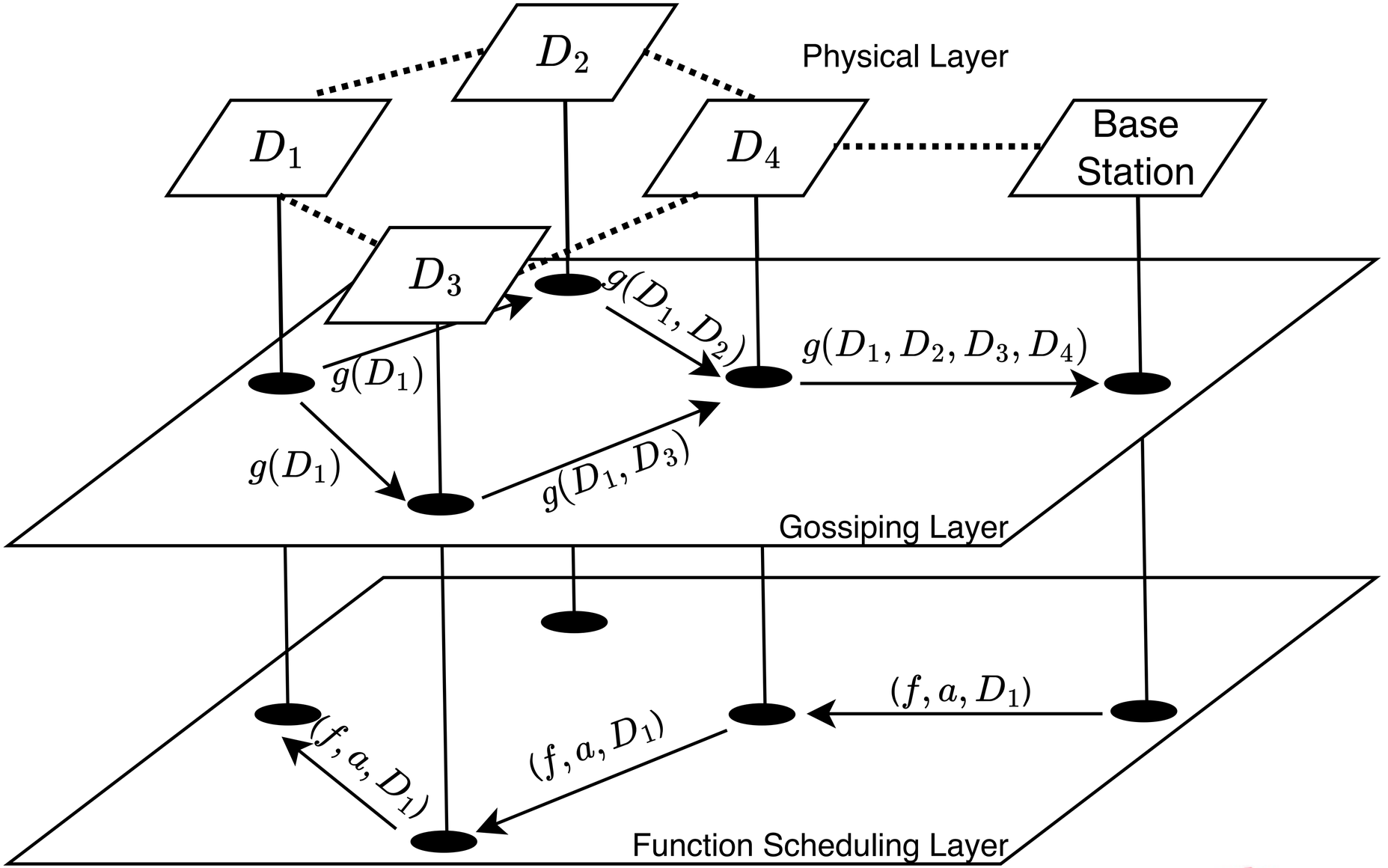
De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2024). **FunLess: Functions-as-a-Service for Private Edge Cloud Systems**. IEEE International Conference on Web Services, **ICWS 2024**, Shenzhen, China, July 7-13, 2024, 961–967. IEEE.

Extending Serverless to Edge Computing

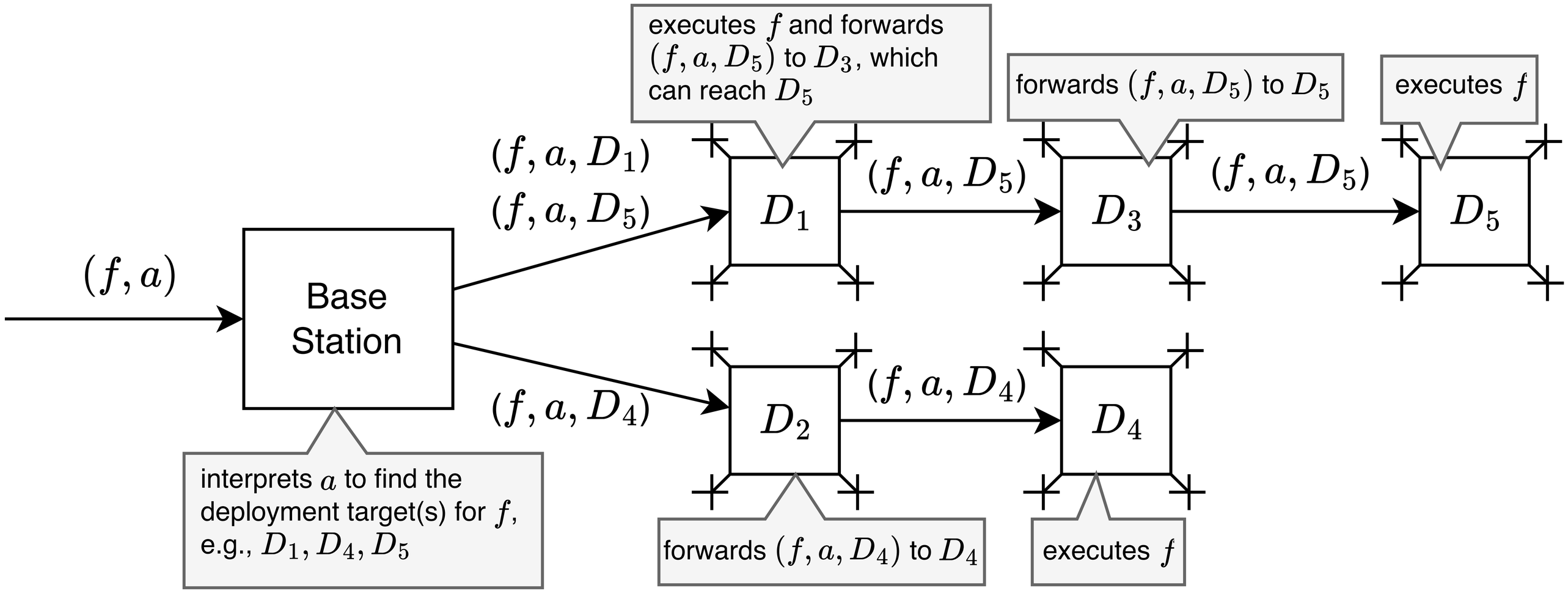


De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2024). **WebAssembly at the Edge: Benchmarking a Serverless Platform for Private Edge Cloud Systems.** IEEE Internet Computing, (01), 1–8.

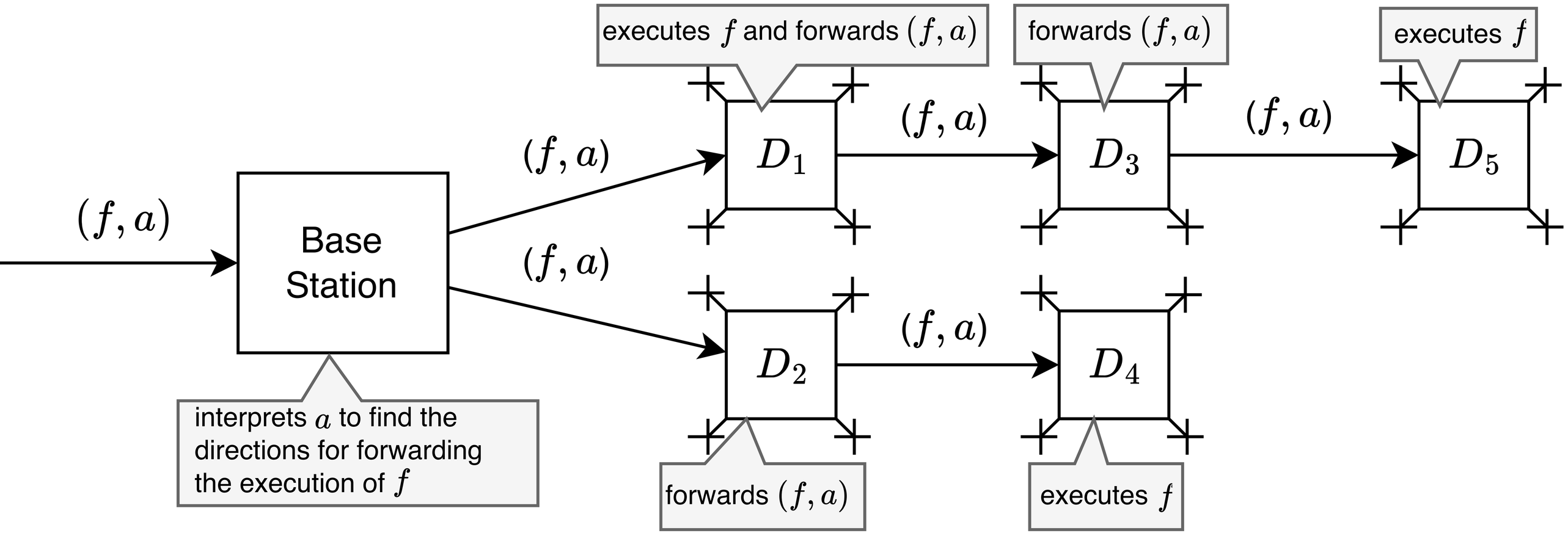
Distributed Serverless Function Scheduling in Ad-Hoc Drone Networks



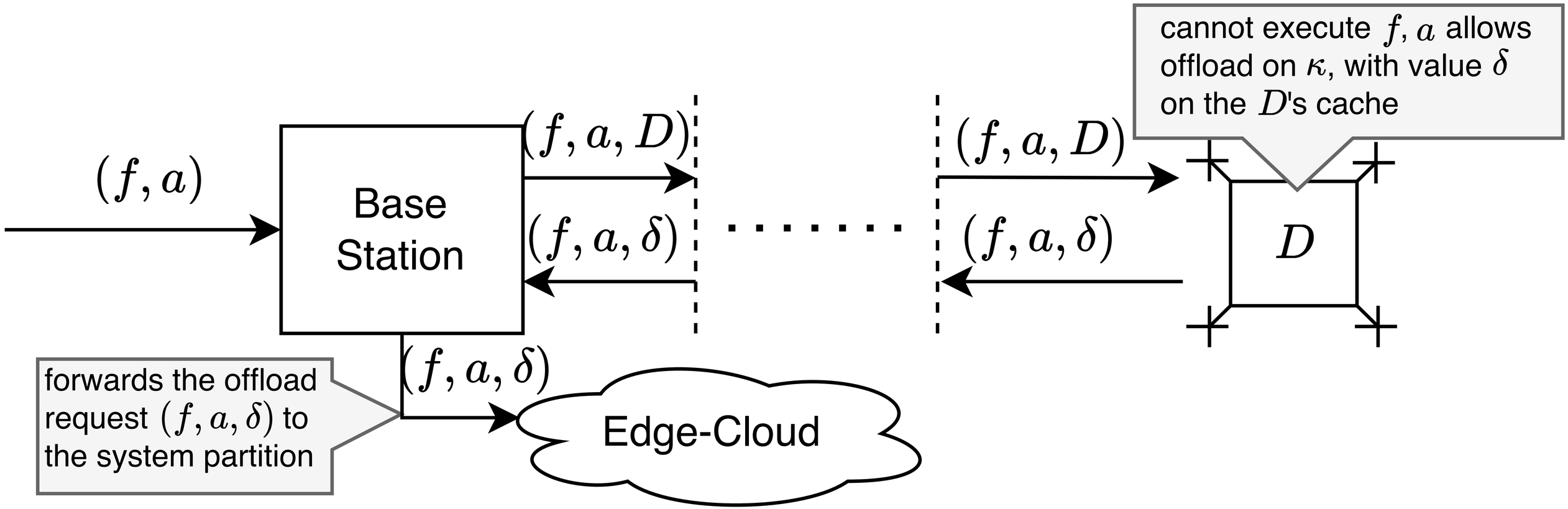
Distributed Serverless Function Scheduling in Ad-Hoc Drone Networks
Hierarchical Scheduling



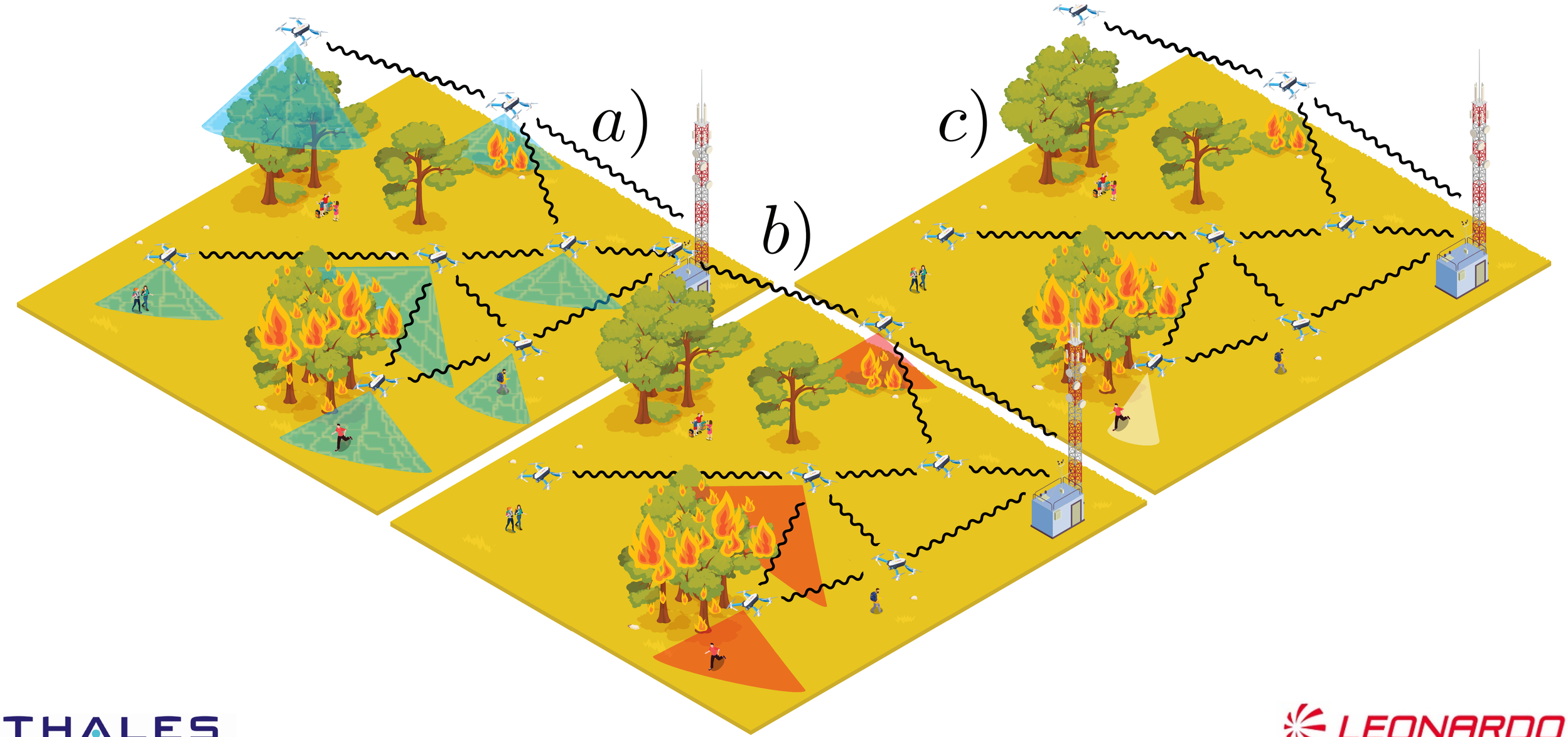
Distributed Serverless Function Scheduling in Ad-Hoc Drone Networks
Progressive Scheduling



Distributed Serverless Function Scheduling in Ad-Hoc Drone Networks Offloading



Distributed Serverless Function Scheduling in Ad-Hoc Drone Networks



THALES

LEONARDO

Distributed Serverless Function Scheduling in Ad-Hoc Drone Networks
AHAPP

Listing 1: `ScanFireAlert` policy.

```
targets: all
attributes:
 device: { camera, gpu }
```

Listing 2: `AnalysePeopleScan` policy.

```
targets: all
affinity: { ScanPeople }
offload: ScanPeopleData
```

Listing 3: `ScanPeople` policy.

```
targets: all
attributes:
 position: { range: 100,
  lat: X, lon: Y, alt: Z }
 device: {
  camera
  gpu
  rotors: { lock: true }
 }
```

THALES

LEONARDO

Distributed Serverless Function Scheduling in Ad-Hoc Drone Networks
AHAPP

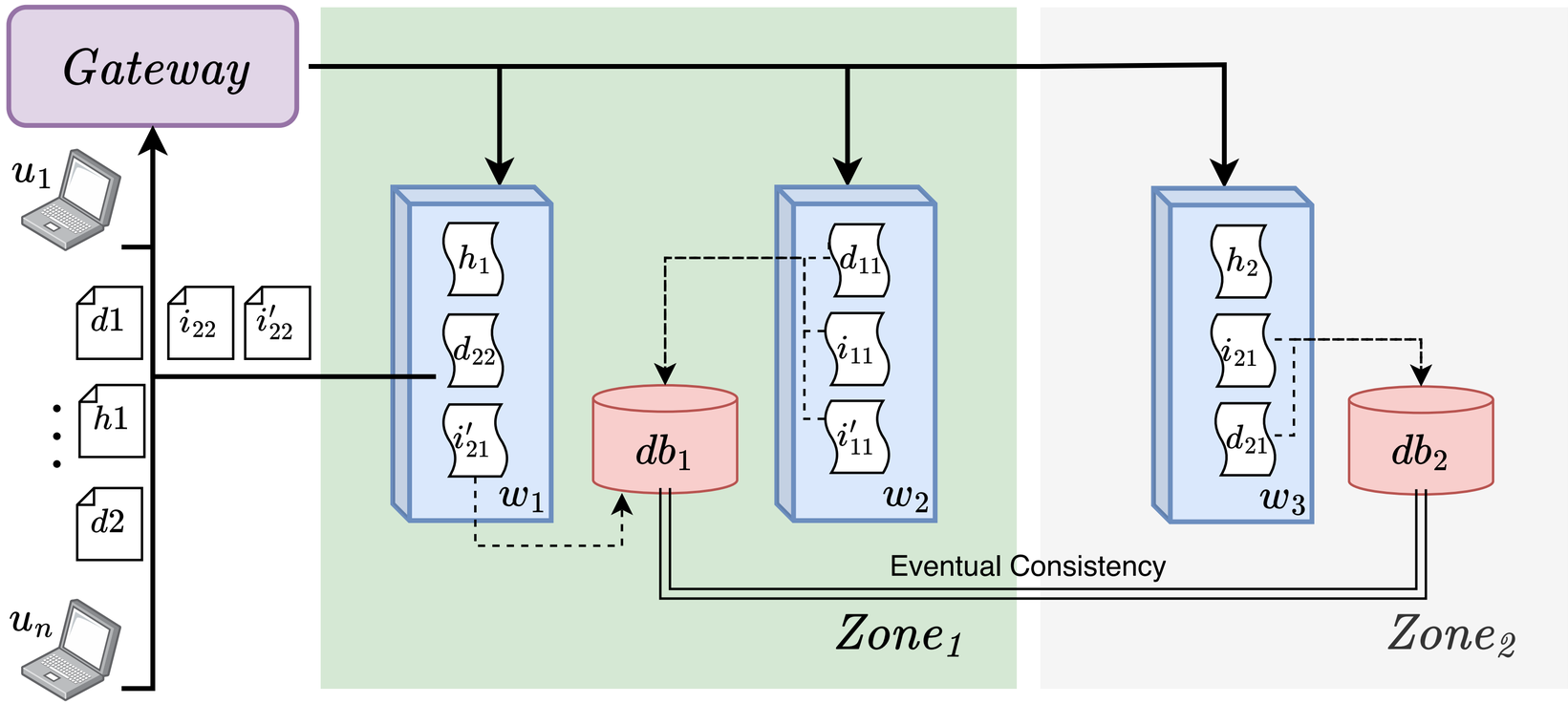
Listing 4: `TrackSurvivor` policy.

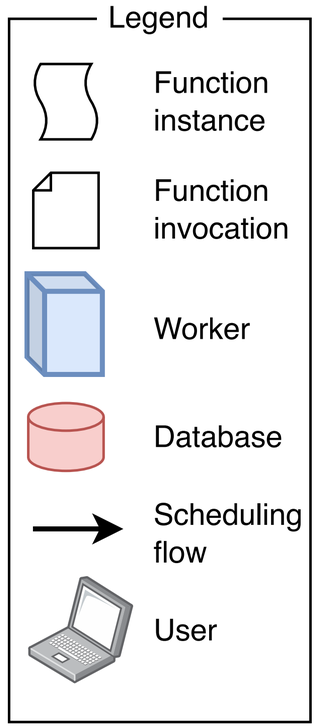
```
targets: one
attributes:
 position: { range: 25,
  lat: X, lon: Y, alt: Z }
 device: {
  camera: { lock: true }
  gpu: { lock: true }
  rotors: { lock: true }
 }
strategy: [ energy, position ]
```

THALES

LEONARDO

Affinity-aware APP



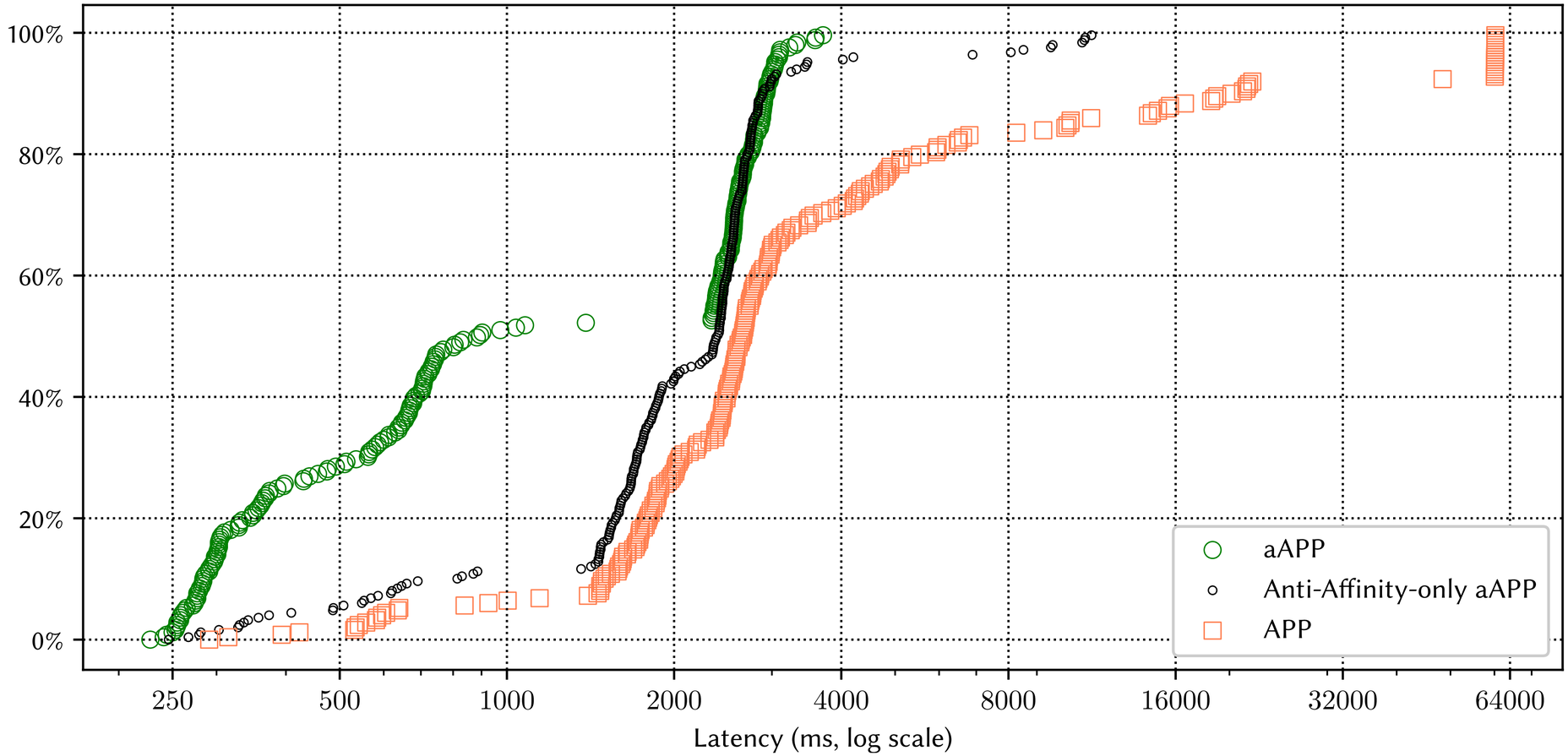



```
- d:
  - workers: *
    affinity:
      - !h
- i:
  - workers: *
    affinity:
      - !h
      - d
- h:
  - workers: *
    affinity:
      - !d
      - !i
```

De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2025). **Affinity-aware Serverless Function Scheduling.** 22nd IEEE International Conference on Software Architecture, **ICSA 2025**, Odense, Denmark, March 31-April 4, 2025. IEEE.

Affinity-aware APP

Configuration	Mean Latency (ms)	Median Latency (ms)	95^{th} Tail Latency (ms)
aAPP	1547	883	3041
anti-affinity-only aAPP	2337 (+40%)	2381 (+91%)	3476 (+13%)
APP	8118 (+135%)	2648 (+99%)	60157 (+180%)



De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2025). **Affinity-aware Serverless Function Scheduling.** 22nd IEEE International Conference on Software Architecture, **ICSA 2025**, Odense, Denmark, March 31-April 4, 2025. IEEE.

Affinity-aware APP • Expressiveness

Deciding the reachability/ co-occurence problem	**APP**
Lower bound	Linear
Upper bound	Linear

Affinity-aware APP • Expressiveness

Deciding the reachability/ co-occurence problem	**APP**	**neg. polarised aAPP**
Lower bound	Linear	Linear
Upper bound	Linear	Linear

Affinity-aware APP • Expressiveness

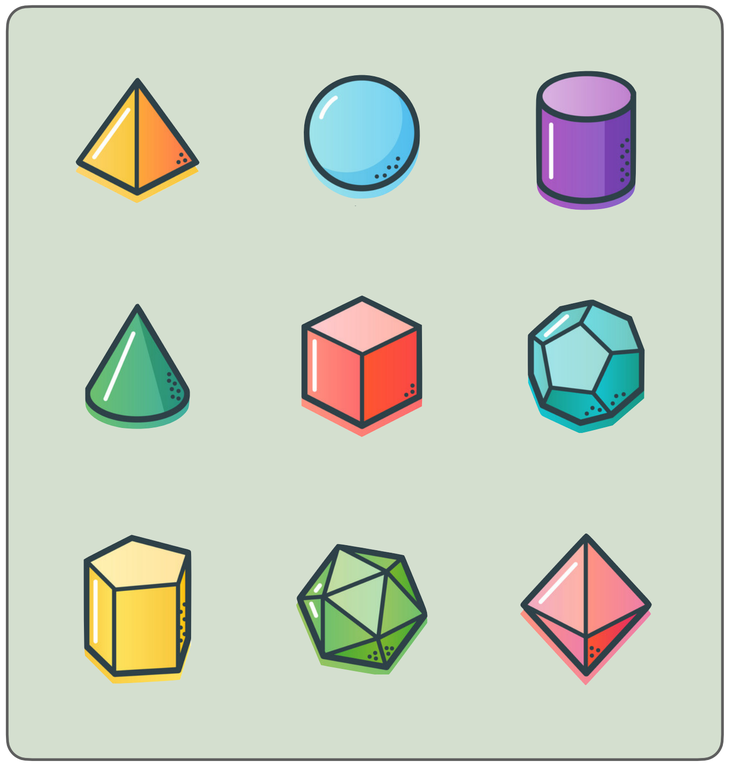
Deciding the reachability/ co-occurence problem	**APP**	**neg. polarised aAPP**	**pos. polarised aAPP**
Lower bound	Linear	Linear	NP (reduction from 3SAT)
Upper bound	Linear	Linear	PSPACE (from aAPP's complexity)

Affinity-aware APP • Expressiveness

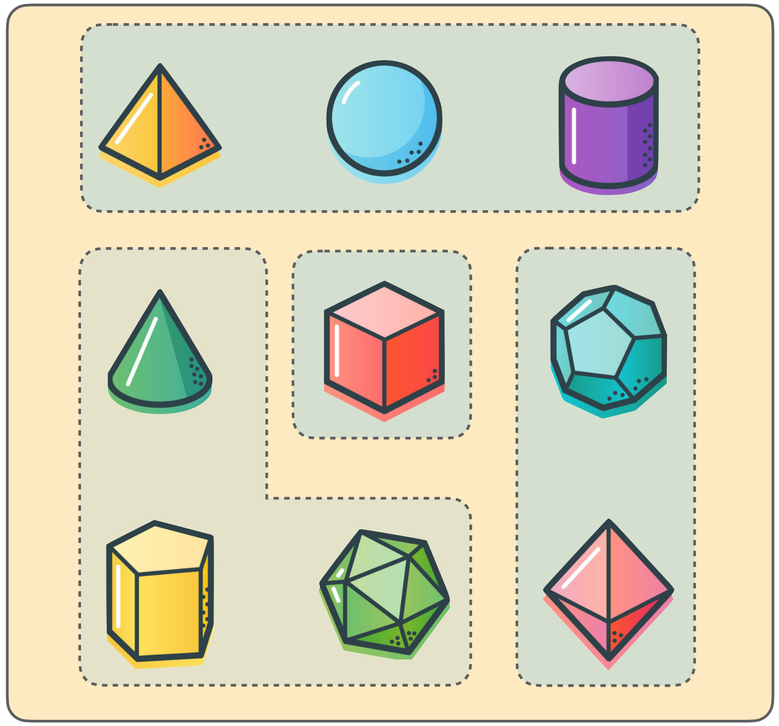
Deciding the reachability/ co-occurence problem	**APP**	**neg. polarised aAPP**	**pos. polarised aAPP**	**aAPP**
Lower bound	Linear	Linear	NP (reduction from 3SAT)	PSPACE (reduction from PLANSAT)
Upper bound	Linear	Linear	PSPACE (from aAPP's complexity)	PSPACE (reduction to PLANSAT)


THANK
YOU

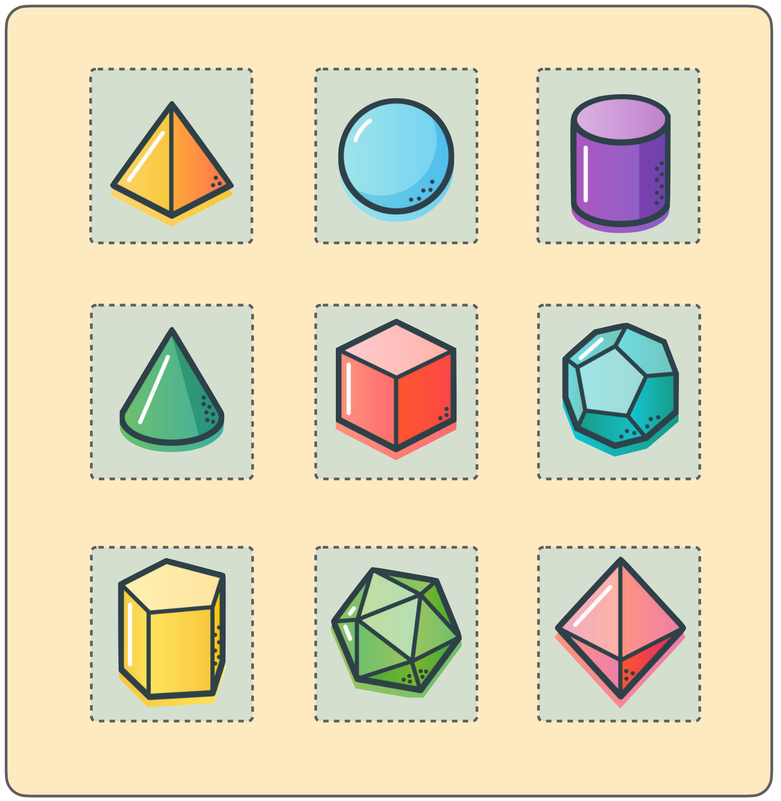
A Gentle Introduction to Serverless



Monolith



Microservices



Serverless

Function

Software Unit

Runtime Environment

Serverless != CGI

Formalising APP's Semantics

Entities

$$f \in \mathcal{F} \qquad w \in \mathcal{W} \subset Identifiers$$

$$t \in \mathcal{T} \subset Identifiers \qquad C \in \mathcal{C} \triangleq \mathcal{W} \to Multiset(\mathcal{F}) \times \mathbb{N} \times \mathbb{N}$$

$$reg \in \mathcal{F} \to \mathbb{N} \times \mathcal{T} \qquad p \in \mathcal{P} \triangleq \mathcal{T} \to List(\mathcal{B})$$

$$[\![\cdot]\!] \ : \ app \to \mathcal{P} \qquad b \in \mathcal{B} \triangleq (List(\mathcal{W}) \cup *) \times s\_opt \times List(i\_opt)$$

$$[C_{start}]\frac{reg(f) = (n,t) \quad p(t) = \bar{b} \quad f, C, \bar{b} \to w \quad w \neq \bot \quad C(w) = (\sigma, n', m)}{C \xRightarrow{(start,f,w)} C[w \mapsto (\sigma \cup \{f\}, n' + n, m)]}$$

$$[C_{fail}]\frac{reg(f) = (\cdot,t) \quad p(t) = \bar{b} \quad f, C, \bar{b} \to \bot}{C \xRightarrow{(fail,f)} C}$$

$$[C_{done}]\frac{w \in \mathrm{dom}(C) \quad f \in \sigma \quad reg(f) = (n,\cdot) \quad C(w) = (\sigma, n', m)}{C \xRightarrow{(done,f,w)} C[w \mapsto (\sigma \setminus \{f\}, n' - n, m)]}$$

De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2024). **Function-as-a-Service Allocation Policies Made Formal**. In T. Margaria & B. Steffen (Eds.), Leveraging Applications of Formal Methods, Verification and Validation. REoCAS Colloquium in Honor of Rocco De Nicola - 12th International Symposium, **ISoLA 2024**, Crete, Greece, October 27-31, 2024, Proceedings, Part I (pp. 306–321). Springer.

Formalising APP's Semantics

Entities

$f \in \mathcal{F}$ $\quad$ $w \in \mathcal{W} \subset Identifiers$

$t \in \mathcal{T} \subset Identifiers$ $\quad$ $C \in \mathcal{C} \triangleq \mathcal{W} \to Multiset(\mathcal{F}) \times \mathbb{N} \times \mathbb{N}$

$reg \in \mathcal{F} \to \mathbb{N} \times \mathcal{T}$ $\quad$ $p \in \mathcal{P} \triangleq \mathcal{T} \to List(\mathcal{B})$

$[\![\cdot]\!] : app \to \mathcal{P}$ $\quad$ $b \in \mathcal{B} \triangleq (List(\mathcal{W}) \cup *) \times s\_opt \times List(i\_opt)$

Blocks Layer

$$[B_{one}]\frac{\overline{w'} = \overline{w} \cap \mathrm{dom}(C) \qquad f, \overline{w'}, s, \overline{i}, C \triangleright w}{f, C, (\overline{w}, s, \overline{i}) \to w} \qquad [B_{star}]\frac{f, \overline{\mathrm{dom}(C)}, s, \overline{i}, C \triangleright w}{f, C, (*, s, \overline{i}) \to w}$$

$$[B_{first}]\frac{f, C, b_1 \to w \qquad w \neq \bot}{f, C, b_1 :: \cdots :: b_n \to w} \qquad [B_{next}]\frac{f, C, b_1 \to \bot \qquad f, C, b_2 :: \cdots :: b_n \to w}{f, C, b_1 :: b_2 :: \cdots :: b_n \to w}$$

Configuration Layer

$$[C_{start}]\frac{reg(f) = (n, t) \qquad p(t) = \overline{b} \qquad f, C, \overline{b} \to w \qquad w \neq \bot \qquad C(w) = (\sigma, n', m)}{C \xRightarrow{(start, f, w)} C[w \mapsto (\sigma \cup \{f\}, n' + n, m)]}$$

$$[C_{fail}]\frac{reg(f) = (\cdot, t) \qquad p(t) = \overline{b} \qquad f, C, \overline{b} \to \bot}{C \xRightarrow{(fail, f)} C}$$

$$[C_{done}]\frac{w \in \mathrm{dom}(C) \qquad f \in \sigma \qquad reg(f) = (n, \cdot) \qquad C(w) = (\sigma, n', m)}{C \xRightarrow{(done, f, w)} C[w \mapsto (\sigma \setminus \{f\}, n' - n, m)]}$$

De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2024). **Function-as-a-Service Allocation Policies Made Formal**. In T. Margaria & B. Steffen (Eds.), Leveraging Applications of Formal Methods, Verification and Validation. REoCAS Colloquium in Honor of Rocco De Nicola - 12th International Symposium, **ISoLA 2024**, Crete, Greece, October 27-31, 2024, Proceedings, Part I (pp. 306–321). Springer.

Formalising APP's Semantics

Workers Layer

$$[W_{first}]\ \frac{\textbf{strategy}(\overline{w}, s) = w \quad w \neq \bot \quad \textbf{valid}(f, w, \overline{i}, C)}{f, \overline{w}, s, \overline{i}, C \triangleright w} \qquad [W_{end}]\ \frac{\textbf{strategy}(\overline{w}, s) = \bot}{f, \overline{w}, s, \overline{i}, C \triangleright \bot}$$

$$[W_{next}]\ \frac{\textbf{strategy}(\overline{w}, s) = w \quad w \neq \bot \quad \neg\textbf{valid}(f, w, \overline{i}, C) \quad f, \overline{w} \setminus w, s, \overline{i}, C \triangleright w'}{f, \overline{w}, s, \overline{i}, C \triangleright w'}$$

Blocks Layer

$$[B_{one}]\ \frac{\overline{w'} = \overline{w} \cap \mathrm{dom}(C) \quad f, \overline{w'}, s, \overline{i}, C \triangleright w}{f, C, (\overline{w}, s, \overline{i}) \to w} \qquad [B_{star}]\ \frac{f, \overline{\mathrm{dom}(C)}, s, \overline{i}, C \triangleright w}{f, C, (*, s, \overline{i}) \to w}$$

$$[B_{first}]\ \frac{f, C, b_1 \to w \quad w \neq \bot}{f, C, b_1 :: \cdots :: b_n \to w} \qquad [B_{next}]\ \frac{f, C, b_1 \to \bot \quad f, C, b_2 :: \cdots :: b_n \to w}{f, C, b_1 :: b_2 :: \cdots :: b_n \to w}$$

Configuration Layer

$$[C_{start}]\ \frac{reg(f) = (n, t) \quad p(t) = \overline{b} \quad f, C, \overline{b} \to w \quad w \neq \bot \quad C(w) = (\sigma, n', m)}{C \xrightarrow{(start, f, w)} C[w \mapsto (\sigma \cup \{f\}, n' + n, m)]}$$

$$[C_{fail}]\ \frac{reg(f) = (\cdot, t) \quad p(t) = \overline{b} \quad f, C, \overline{b} \to \bot}{C \xrightarrow{(fail, f)} C}$$

$$[C_{done}]\ \frac{w \in \mathrm{dom}(C) \quad f \in \sigma \quad reg(f) = (n, \cdot) \quad C(w) = (\sigma, n', m)}{C \xrightarrow{(done, f, w)} C[w \mapsto (\sigma \setminus \{f\}, n' - n, m)]}$$

Entities

$f \in \mathcal{F}$ — $w \in \mathcal{W} \subset Identifiers$

$t \in \mathcal{T} \subset Identifiers$ — $C \in \mathcal{C} \triangleq \mathcal{W} \to Multiset(\mathcal{F}) \times \mathbb{N} \times \mathbb{N}$

$reg \in \mathcal{F} \to \mathbb{N} \times \mathcal{T}$ — $p \in \mathcal{P} \triangleq \mathcal{T} \to List(\mathcal{B})$

$[\![\cdot]\!] : app \to \mathcal{P}$ — $b \in \mathcal{B} \triangleq (List(\mathcal{W}) \cup *) \times s\_opt \times List(i\_opt)$

strategy relation and valid predicate

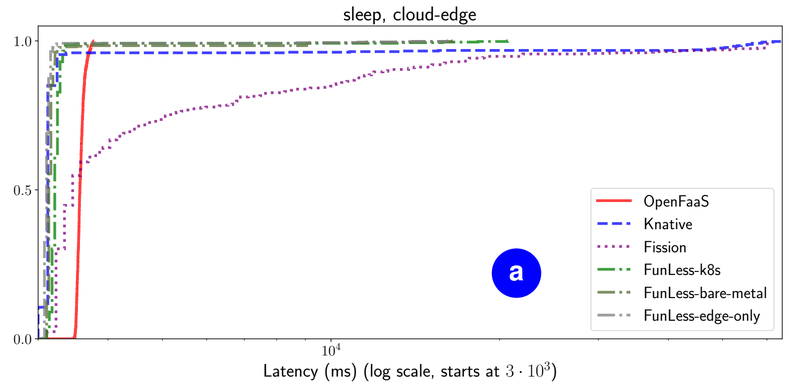
$$\textbf{strategy}(\overline{w}, s) = \begin{cases} w & \text{if } s = \texttt{any} \wedge w \in \{\overline{w}\} \\ w & \text{if } s = \texttt{best\_first} \wedge \overline{w} = w :: \overline{w'} \\ \bot & \text{otherwise} \end{cases}$$

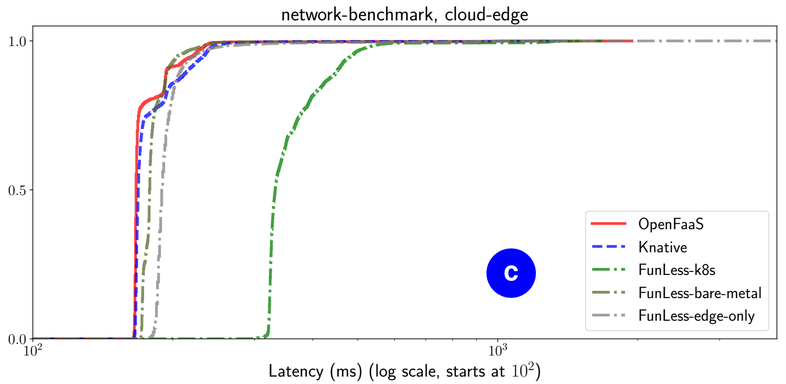
$$\textbf{valid}(f, w, i_1 :: \cdots :: i_n, C) = \textbf{valid}(f, w, i_1, C) \ \wedge \ \cdots \ \wedge \ \textbf{valid}(f, w, i_n, C)$$

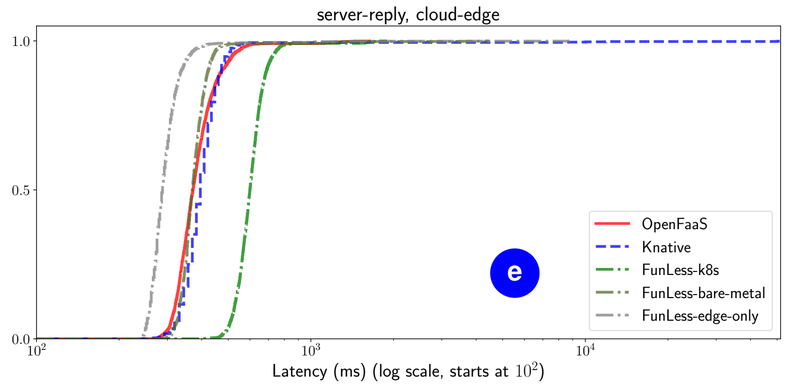
$$\textbf{valid}(f, w, i, C) = \begin{cases} \texttt{true} & \text{if } i = \texttt{capacity\_used } n\texttt{\%} \\ & \quad \wedge\ reg(f) = (n_f, \cdot) \wedge\ C(w) = (\cdot, n_{cur}, n_{max}) \\ & \quad \wedge\ \min(n, 100) \geq 100 * (n_{cur} + n_f)/n_{max} \\ \texttt{true} & \text{if } i = \texttt{max\_concurrent\_invocations } n \\ & \quad \wedge\ \textbf{valid}(f, w, \texttt{capacity\_used } 100\texttt{\%}, C) \\ & \quad \wedge\ C(w) = (\sigma, \cdot, \cdot) \wedge\ n \geq |\sigma| + 1 \\ \texttt{false} & \text{otherwise} \end{cases}$$

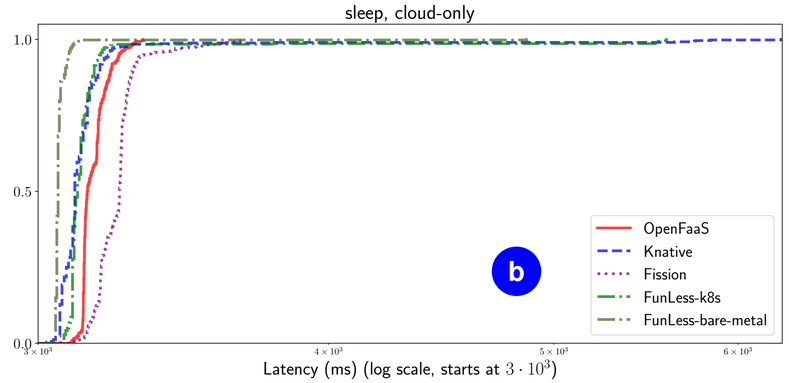
De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2024). **Function-as-a-Service Allocation Policies Made Formal**. In T. Margaria & B. Steffen (Eds.), Leveraging Applications of Formal Methods, Verification and Validation. REoCAS Colloquium in Honor of Rocco De Nicola - 12th International Symposium, **ISoLA 2024**, Crete, Greece, October 27-31, 2024, Proceedings, Part I (pp. 306–321). Springer.

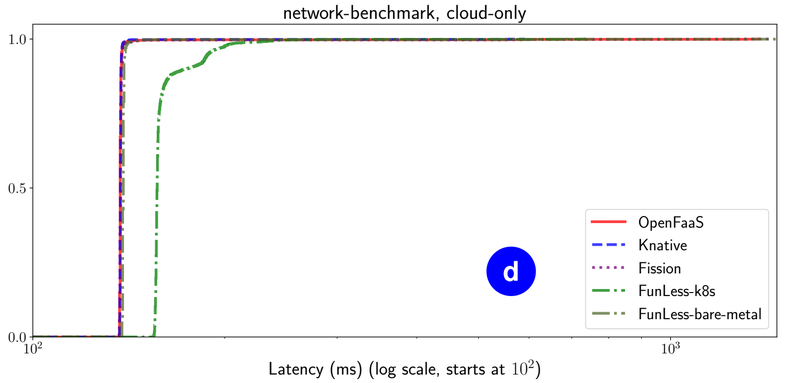
Extending Serverless to Edge Computing

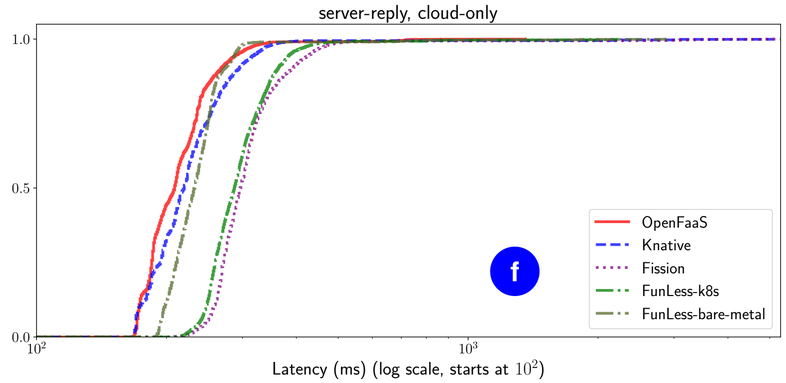


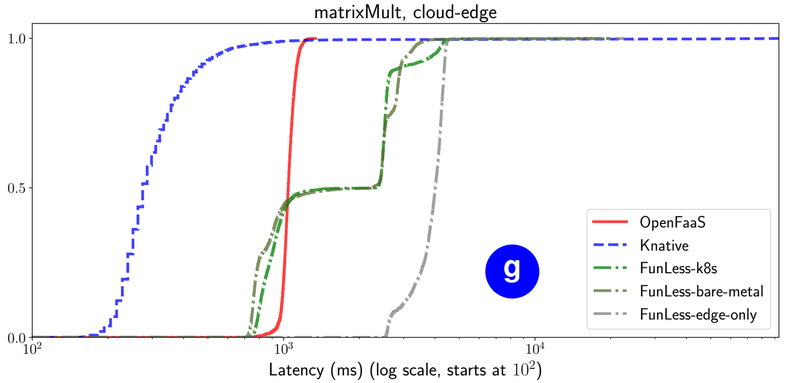


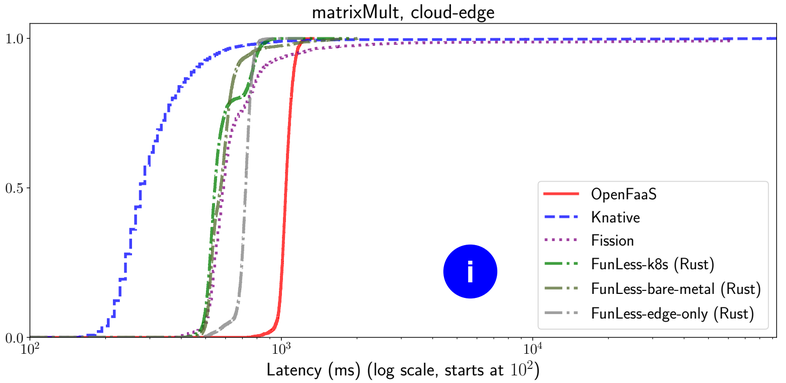












De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2024). **WebAssembly at the Edge: Benchmarking a Serverless Platform for Private Edge Cloud Systems.** IEEE Internet Computing, (01), 1–8.

Cost-aware APP

```
1   // tag: mapReduce
2   ( jobs, m, r ) => {
3     for(i in range(0, m)) {
4       call Map(jobs, i)
5       for(j in range(0, r)) {
6         call Reduce(jobs, i, j)
7       }
8     }
9   }
10
```

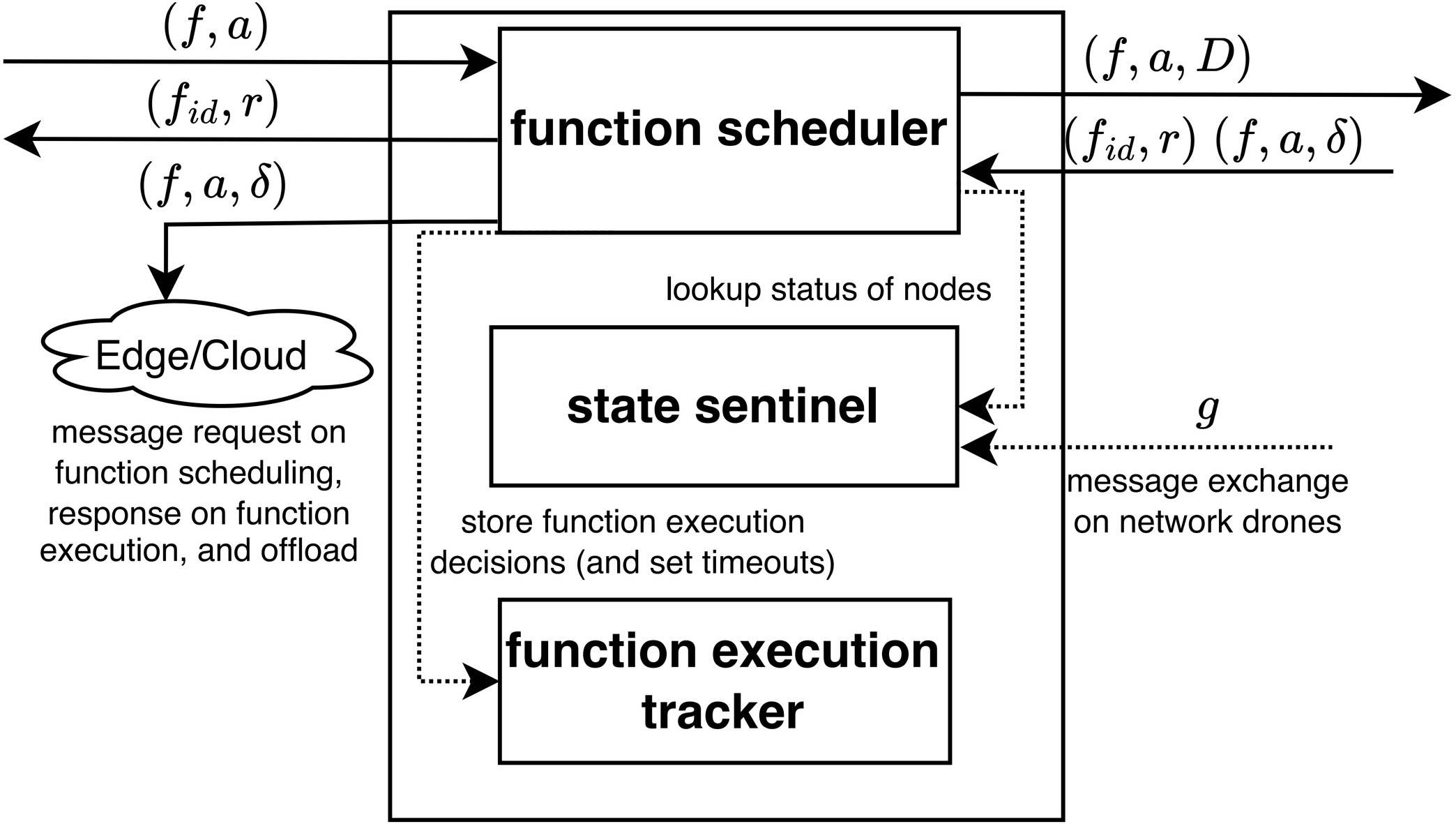
$$
\begin{array}{lll}
main(J,m,r,M,R) = for_2(0,m,r,M,R) & & [\,] \\
for_2(i,m,r,M,R) = & M + for_4(0,r,R) + for_2(i+1,m,r,M,R) & [\,m \geq i\,] \\
for_2(i,m,r,M,R) = & 0 & [\,i \geq m+1\,] \\
for_4(j,r,R) = & R + for_4(j+1,r,R) & [\,r \geq j\,] \\
for_4(j,r,R) = & 0 & [\,j \geq r+1\,]
\end{array}
$$

```
- mapReduce:
 - workers:
    - wrk: W1
    - wrk: W2
   invalidate: max_latency
```

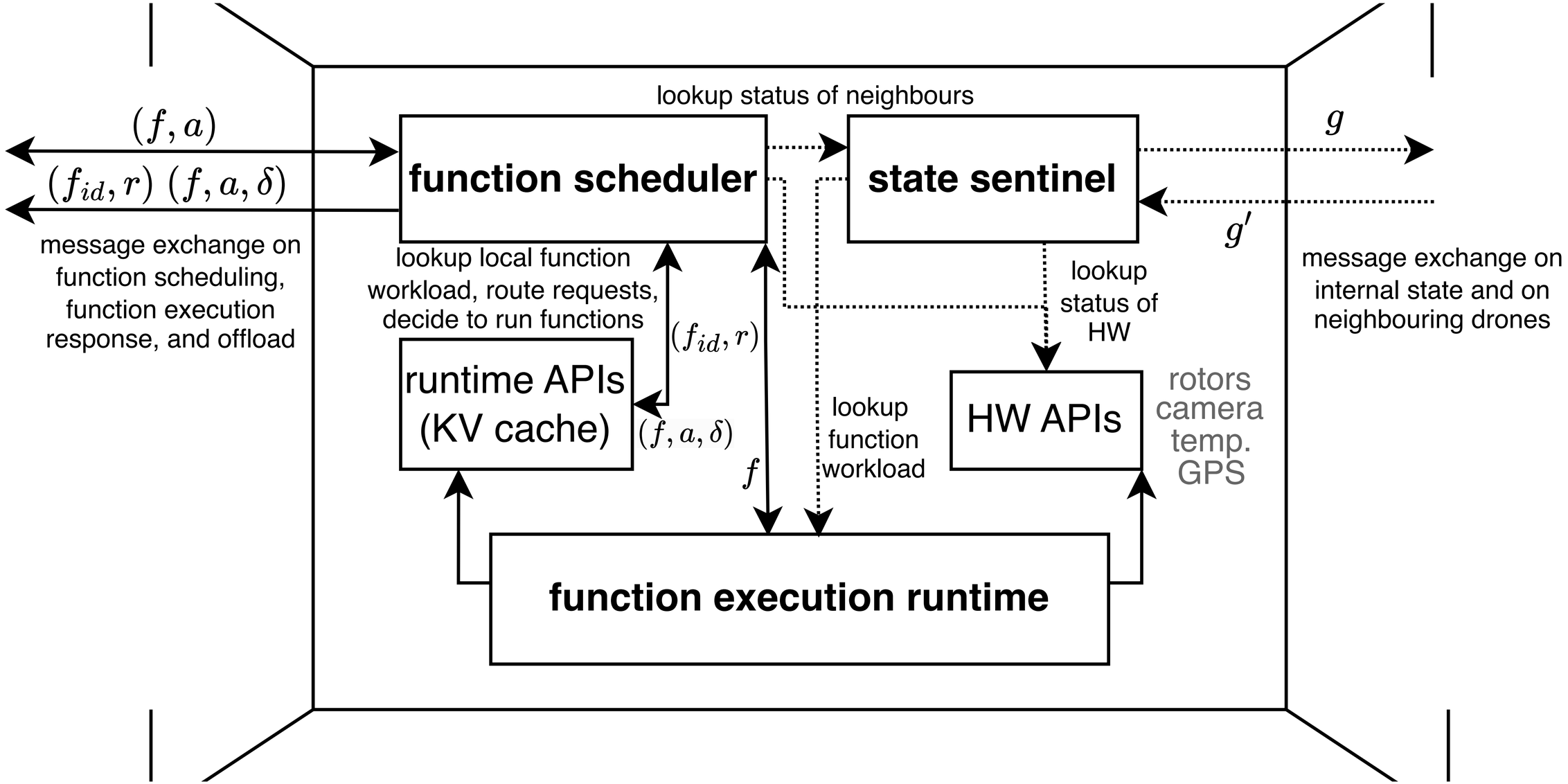
Cost Expression: m*(M + r*R)

```
W in ( W1, W2 )
 where m *( W.latency( Map )
          + r * W.latency( Reduce ) )
 is < 300
```

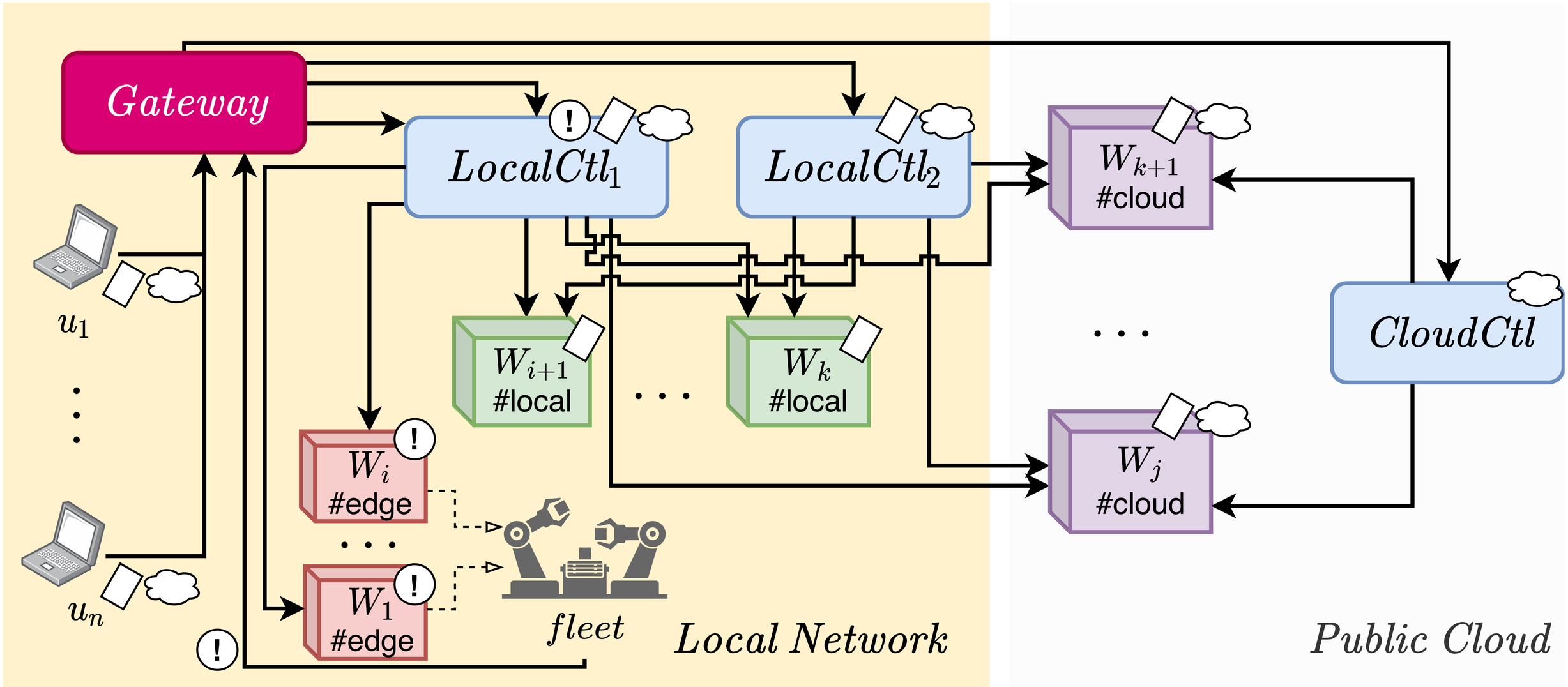
Distributed Serverless Function Scheduling in Ad-Hoc Drone Networks
Software Components • Base Station



Distributed Serverless Function Scheduling in Ad-Hoc Drone Networks
Software Components • Node



Topology-aware APP



De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2022). **A Declarative Approach to Topology-Aware Serverless Function-Execution Scheduling**. In C. A. Ardagna, … J. Zhang (Eds.), IEEE International Conference on Web Services, **ICWS 2022**, Barcelona, Spain, July 10-16, 2022 (pp. 337–342). IEEE.

Topology-aware APP

```
critical:
 - controller: LocalCtl_1
   workers:
    - *edge
   strategy: random
 followup: fail
machine_learning:
 - controller: CloudCtl
   workers:
    - *cloud
   topology_tolerance: same
 followup: default
```

```
default:
 - controller: LocalCtl_1
   workers:
    - *internal
      strategy: random
    - *cloud
      strategy: random
   strategy: best_first
 - controller: LocalCtl_2
   workers: # same as above
   strategy: best_first
 strategy: random
```

De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2022). **A Declarative Approach to Topology-Aware Serverless Function-Execution Scheduling**. In C. A. Ardagna, … J. Zhang (Eds.), IEEE International Conference on Web Services, **ICWS 2022**, Barcelona, Spain, July 10-16, 2022 (pp. 337–342). IEEE.