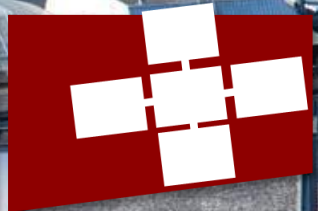
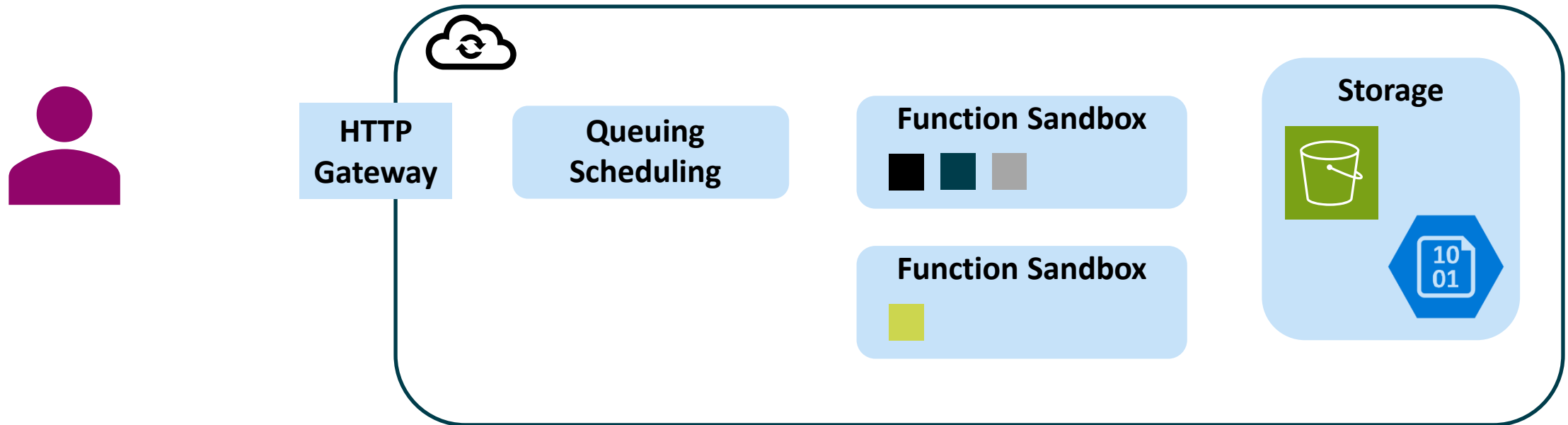


MARCIN COPIK, ALEXANDRU CALOTOIU, GYORGY RETHY, ROMAN BÖHRINGER, RODRIGO BRUNO, TORSTEN HOEFLER

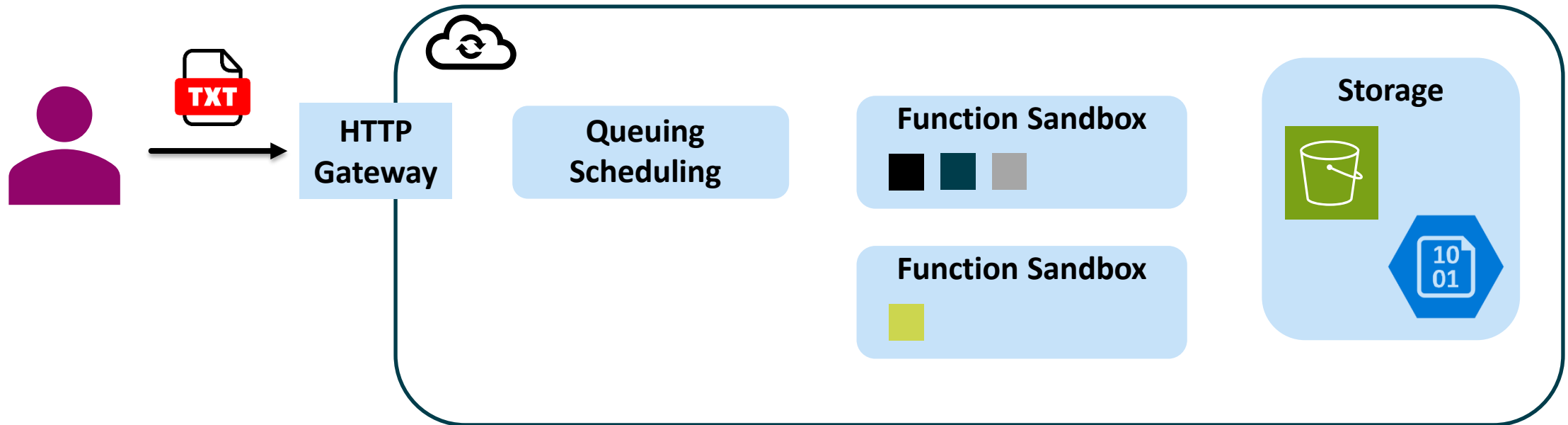
Process-as-a-Service: Unifying Elastic and Stateful Clouds with Serverless Processes



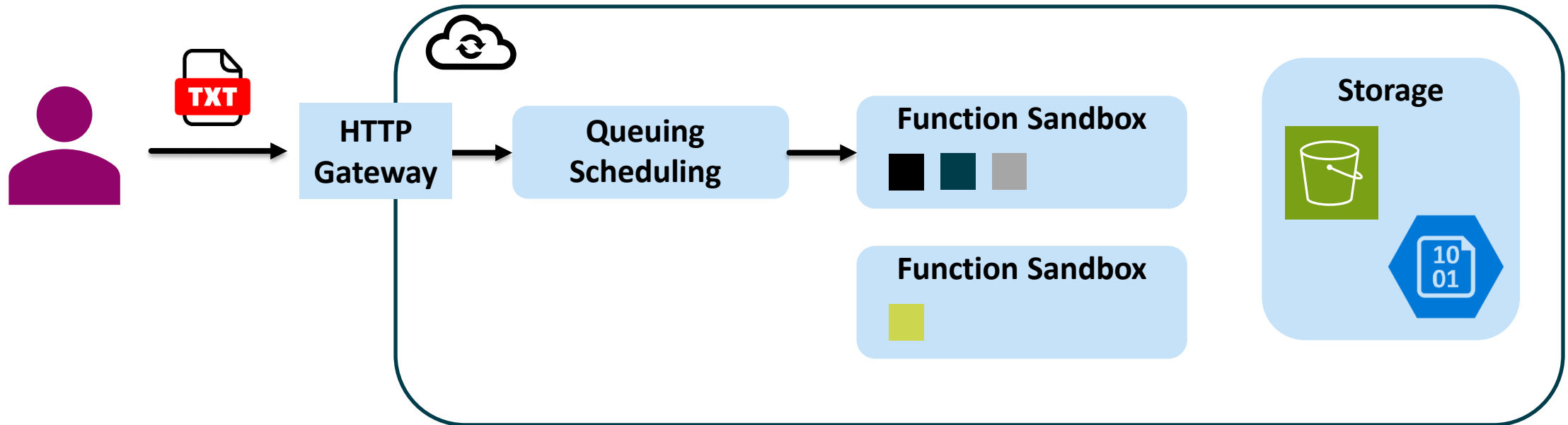
LaTeX Microservice in Serverless



LaTeX Microservice in Serverless

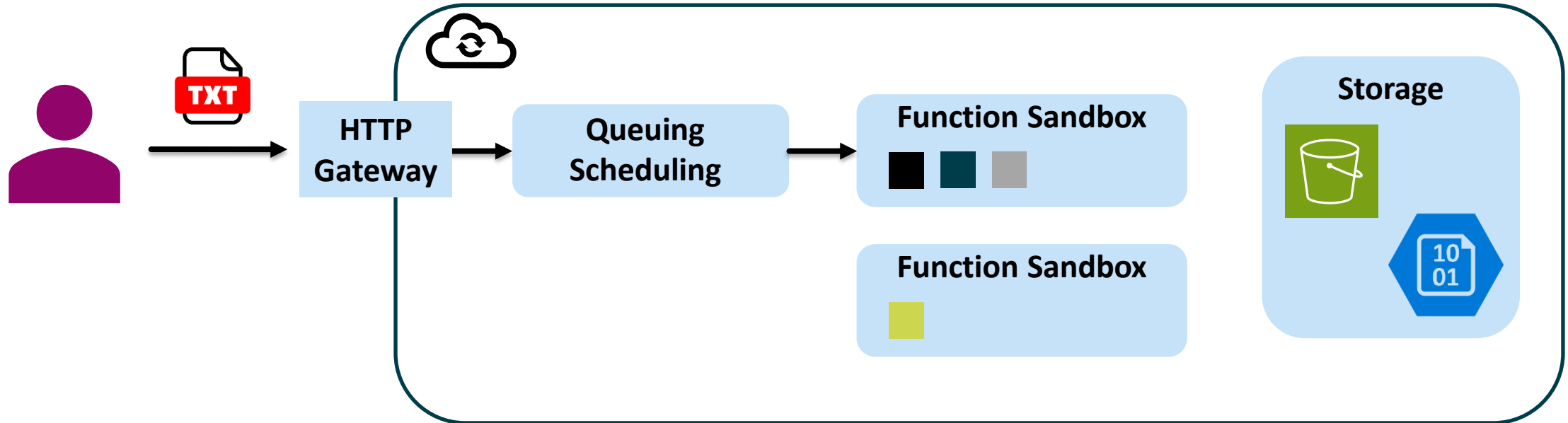


LaTeX Microservice in Serverless



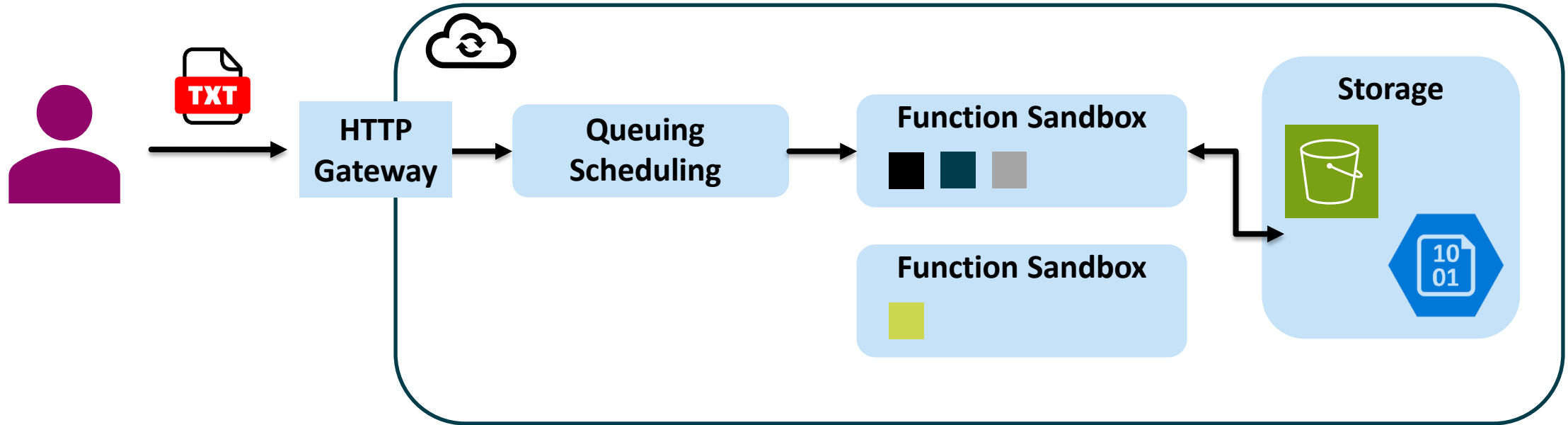
LaTeX Microservice in Serverless

Is cache up to date?
 Did other function edit file of this client?

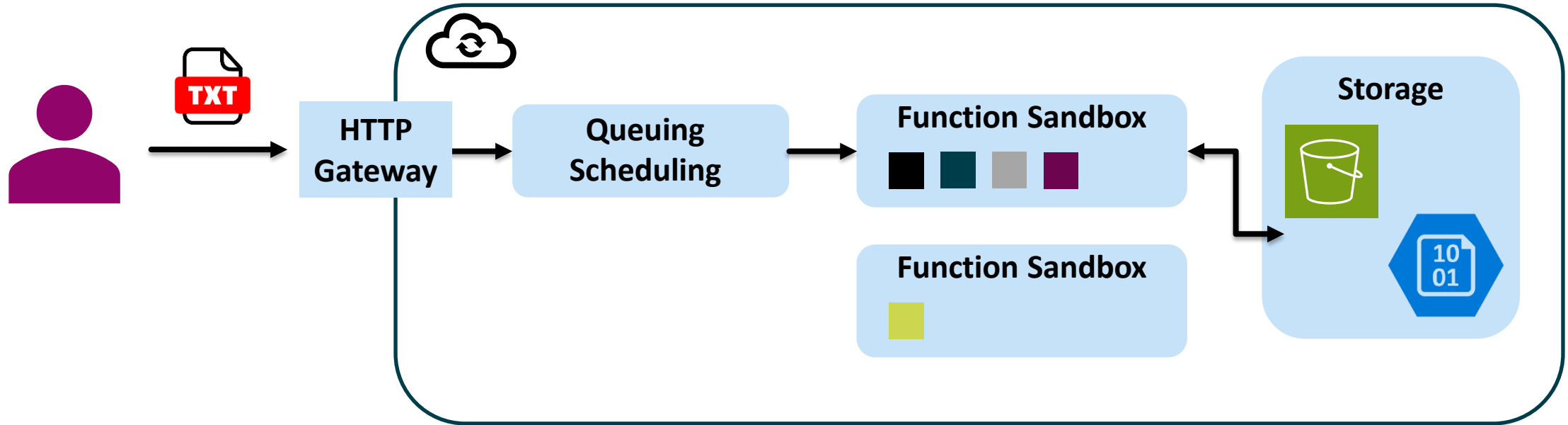


LaTeX Microservice in Serverless

Is cache up to date?
 Did other function edit file of this client?



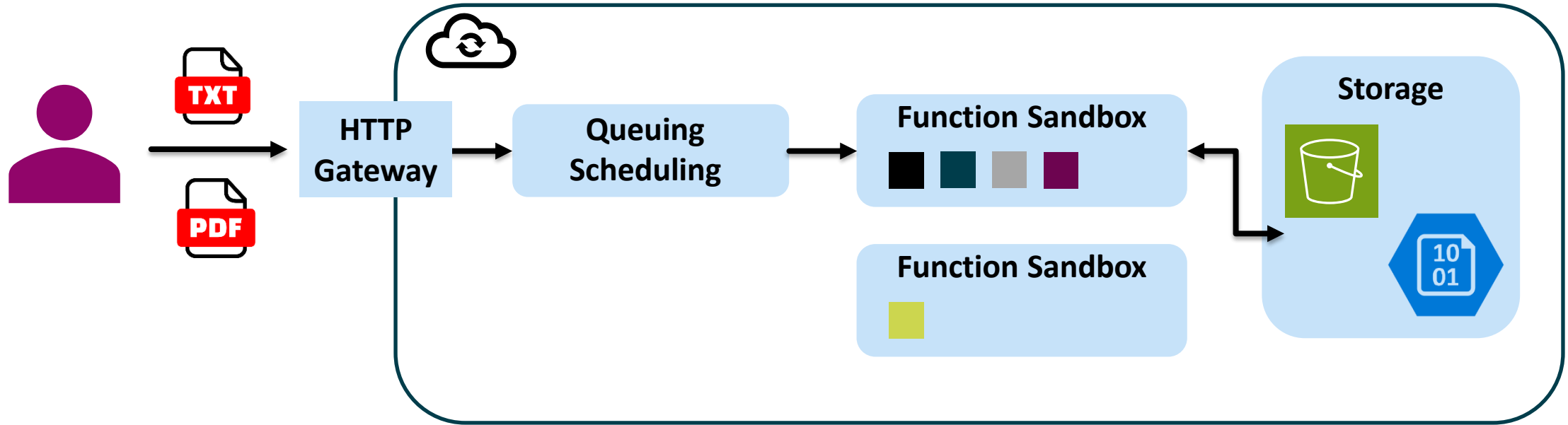
LaTeX Microservice in Serverless



Serverless Service


 Cached function state is not reliable

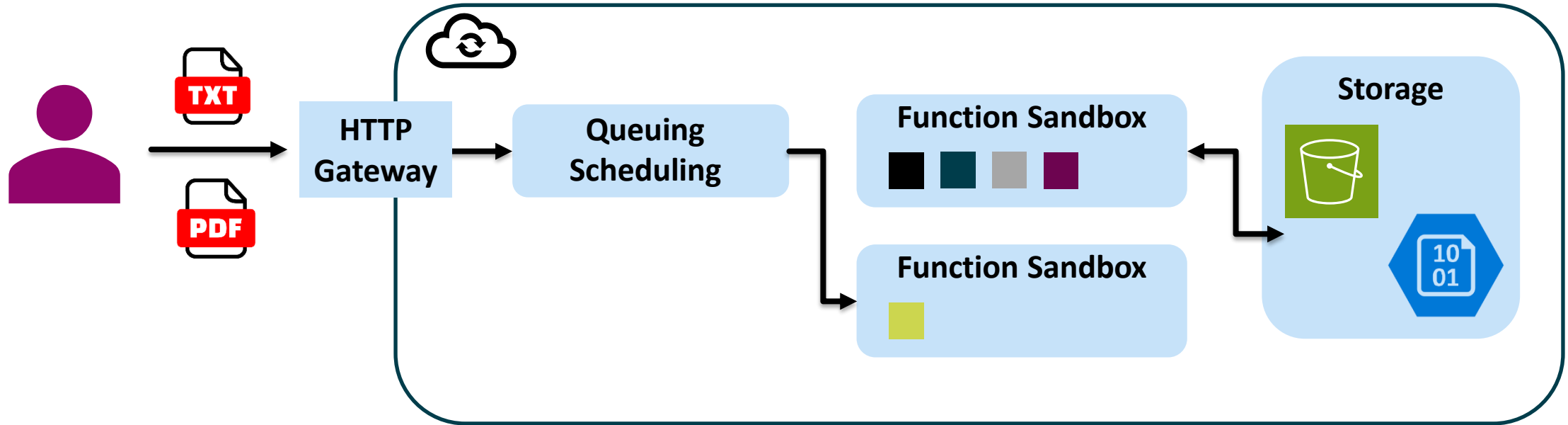
LaTeX Microservice in Serverless



Serverless Service


 Cached function state is not reliable

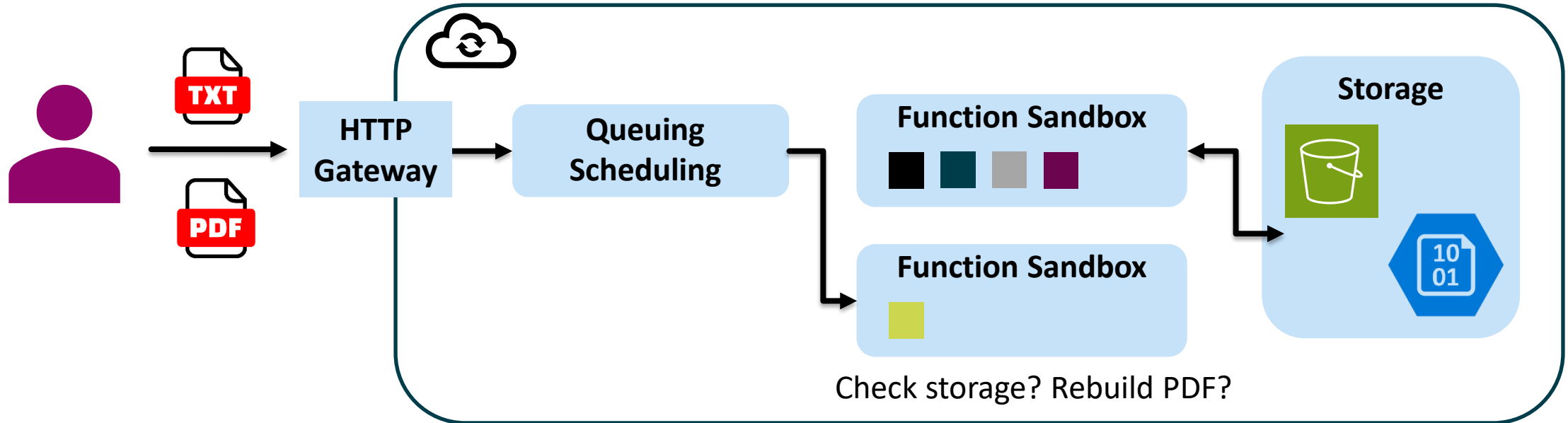
LaTeX Microservice in Serverless



Serverless Service


 Cached function state is not reliable

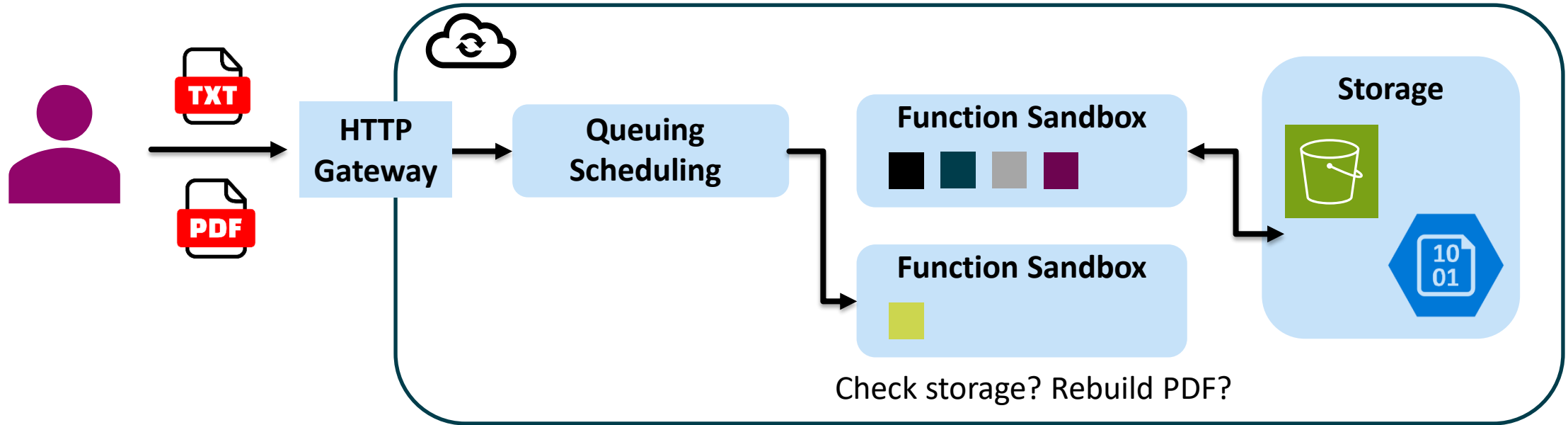
LaTeX Microservice in Serverless



Serverless Service

✘ Cached function state is not reliable

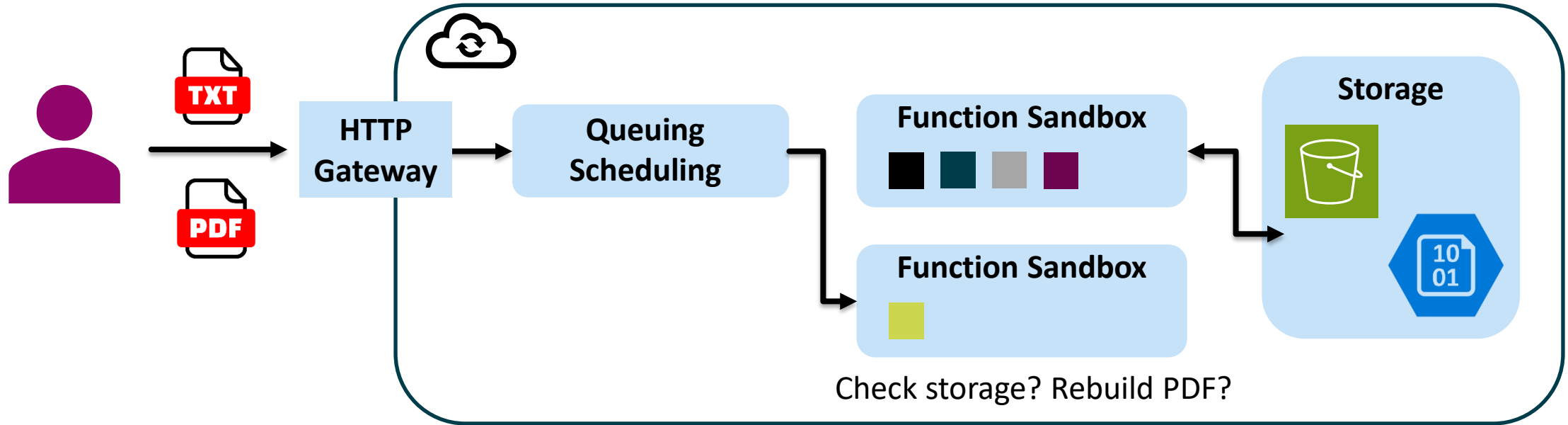
LaTeX Microservice in Serverless



Serverless Service

- ✘ Cached function state is not reliable
- ✘ Request context has no influence on scheduling

LaTeX Microservice in Serverless



Serverless Service

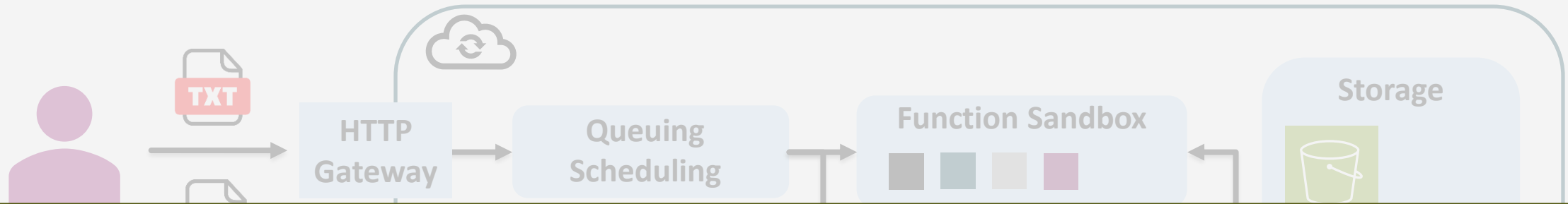
- ✗ Cached function state is not reliable
- ✗ Request context has no influence on scheduling

Standard Microservice

- ✓ Extensive caching
- ✓ Load balancing with sticky sessions

LaTeX Microservice in Serverless

Is cache up to date?
Did other function edit file of this client?



Should we fix this problem with dedicated optimizations?

Maybe functions are not enough and need a better abstraction?

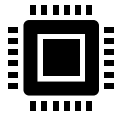
Serverless Service

- ✗ Cached function state is not reliable
- ✗ Request context has no influence on scheduling

Standard Microservice

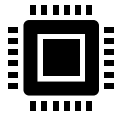
- ✓ Extensive caching
- ✓ Load balancing with sticky sessions

Cloud Models



Cloud Models

IaaS



Persistent



Provisioned



Persistent,
local



Months

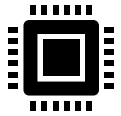


Manual,
minutes

Cloud Models

IaaS

CaaS



Persistent

Persistent



Provisioned

Provisioned



Persistent,
local

Persistent,
local



Months

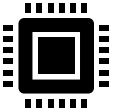




Hours, days



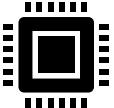




Manual,
minutes

Semi-automatic,
minutes & seconds

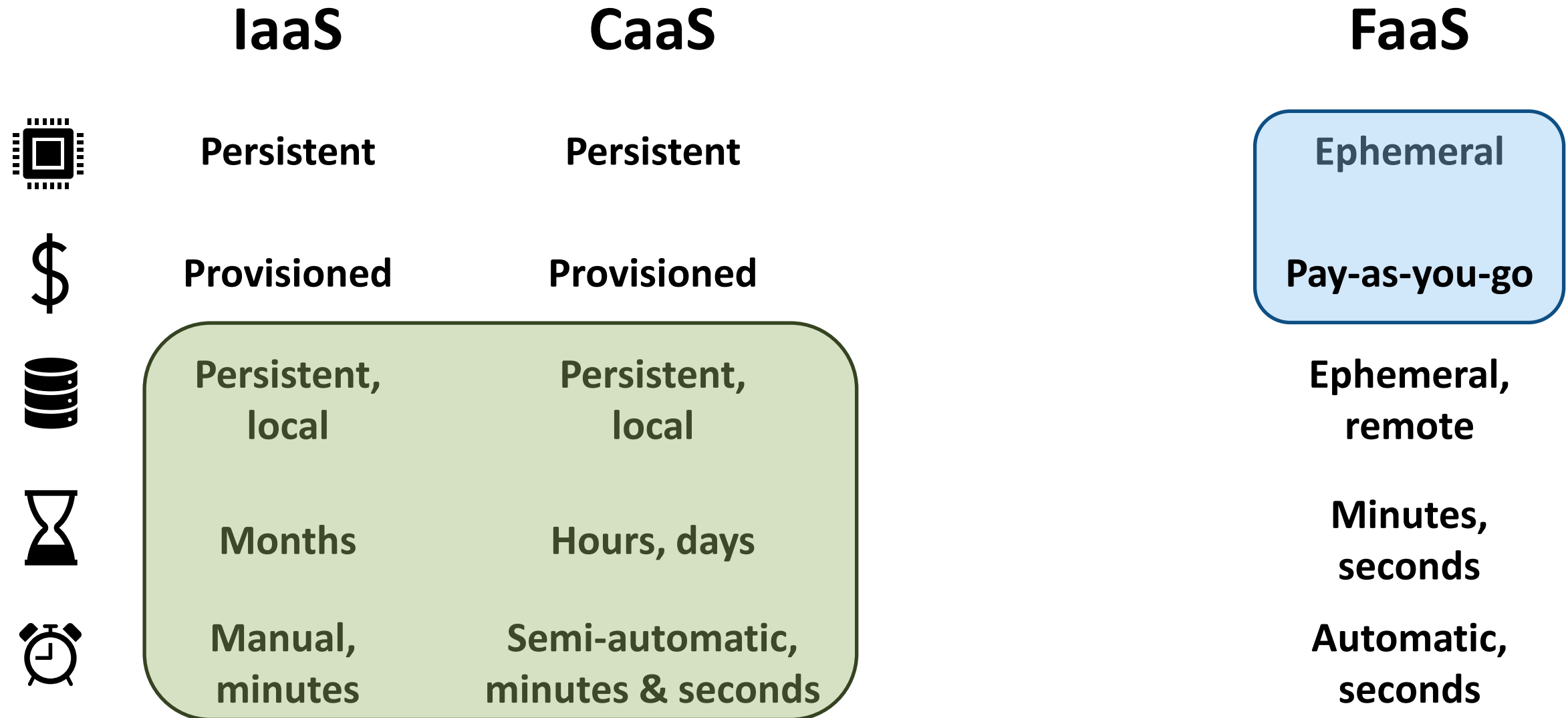
Cloud Models

	IaaS	CaaS	FaaS
	Persistent	Persistent	Ephemeral
	Provisioned	Provisioned	Pay-as-you-go
	Persistent, local	Persistent, local	Ephemeral, remote
	Months	Hours, days	Minutes, seconds
	Manual, minutes	Semi-automatic, minutes & seconds	Automatic, seconds

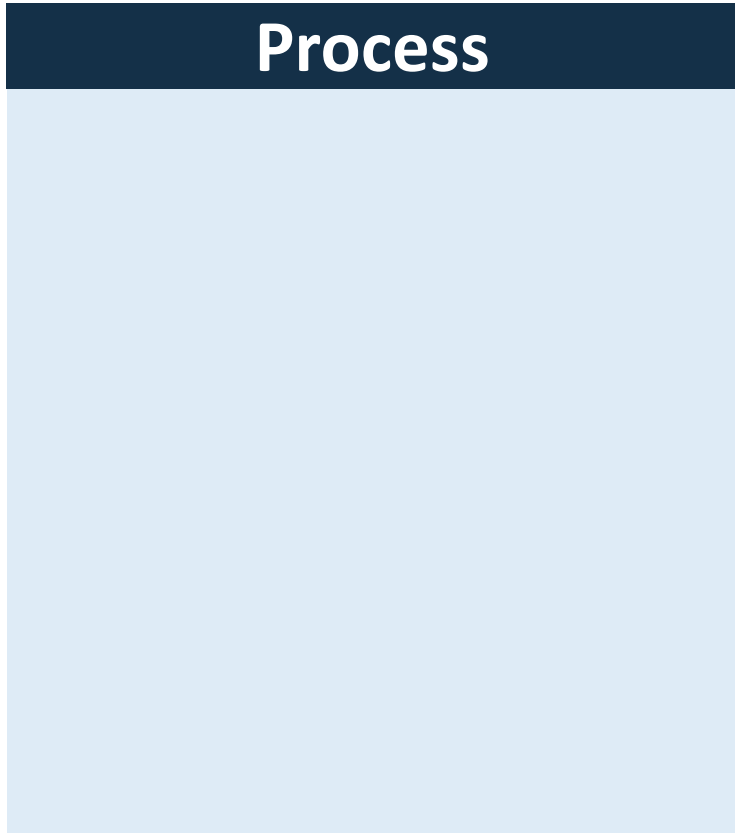
Cloud Models

	IaaS	CaaS	FaaS
	Persistent	Persistent	Ephemeral
	Provisioned	Provisioned	Pay-as-you-go
	Persistent, local	Persistent, local	Ephemeral, remote
	Months	Hours, days	Minutes, seconds
	Manual, minutes	Semi-automatic, minutes & seconds	Automatic, seconds

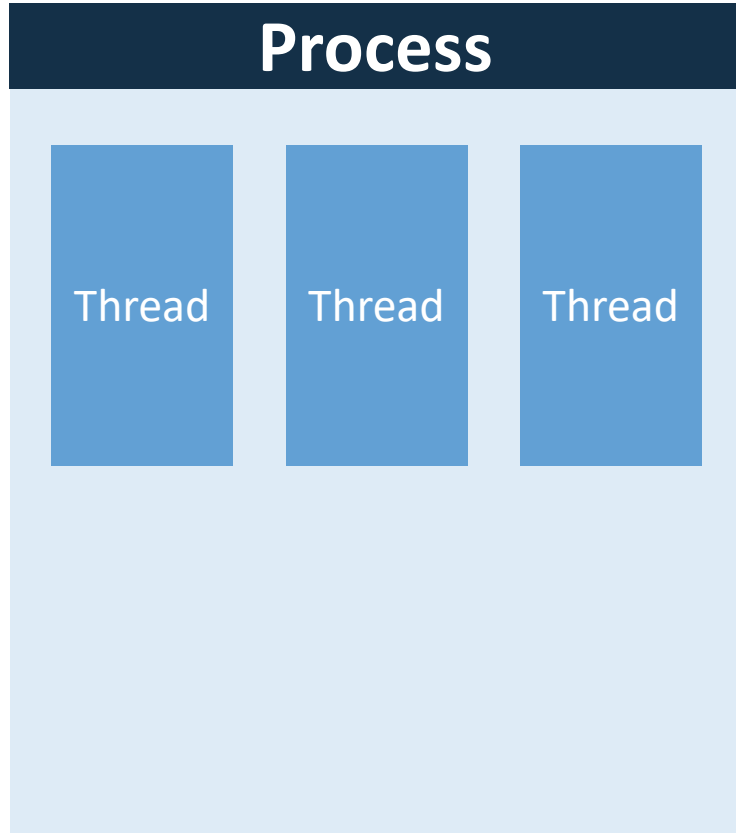
Cloud Models



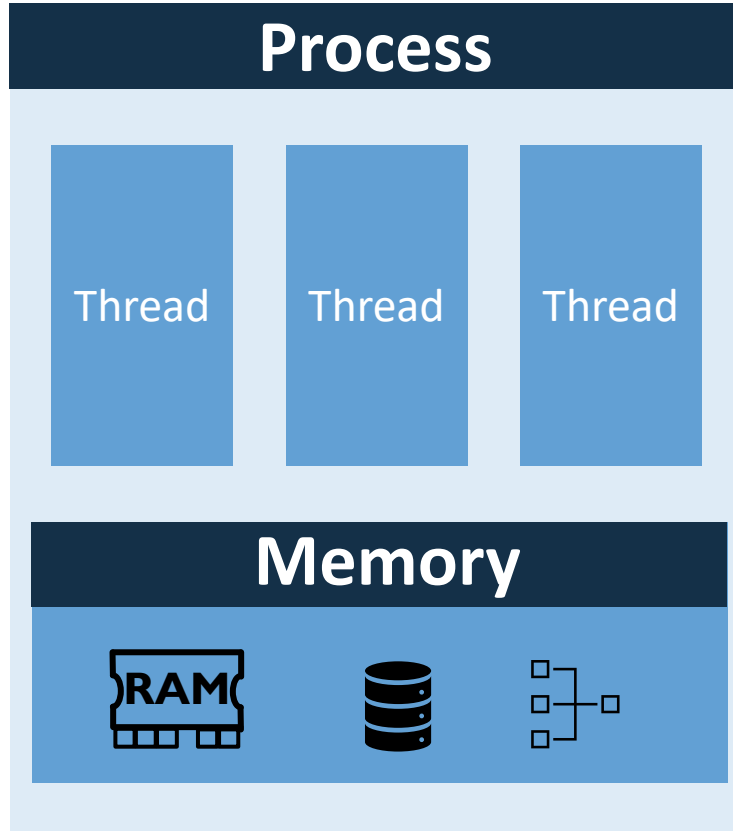
POSIX Process Model



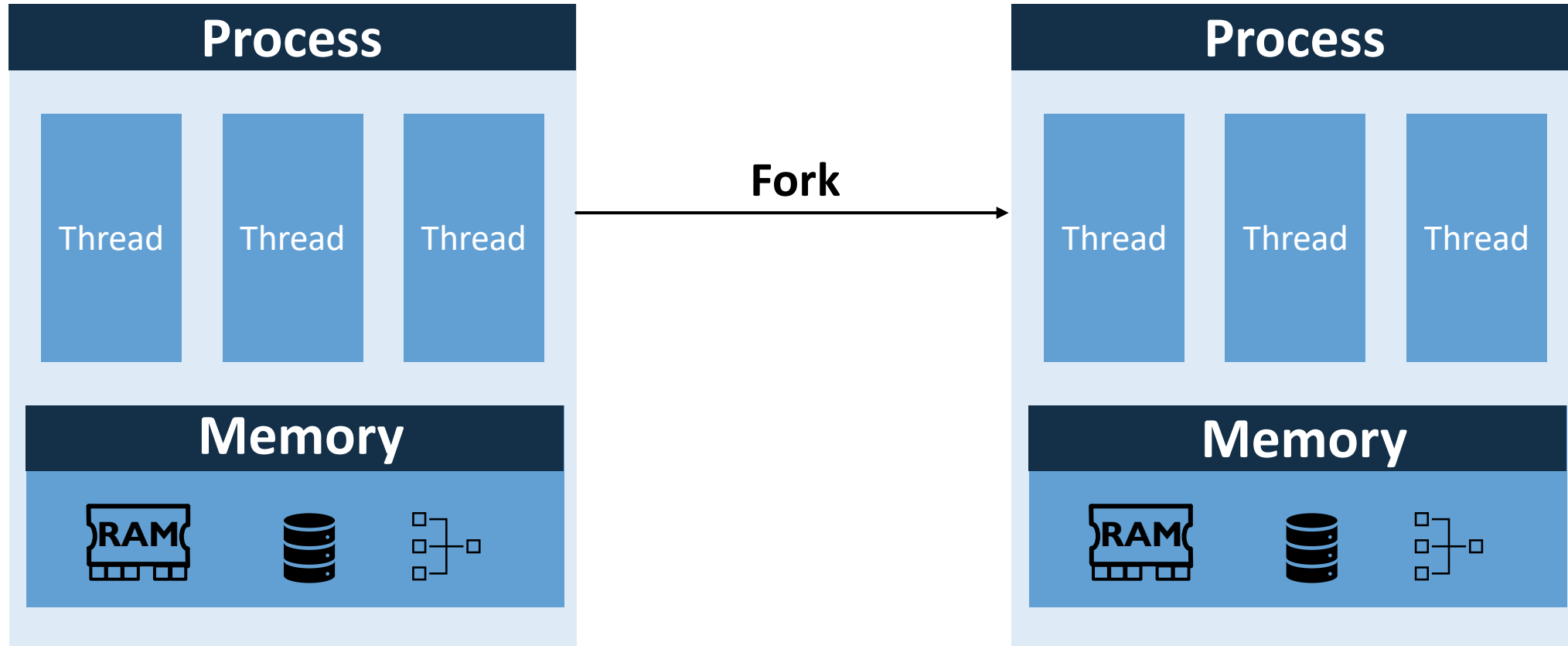
POSIX Process Model



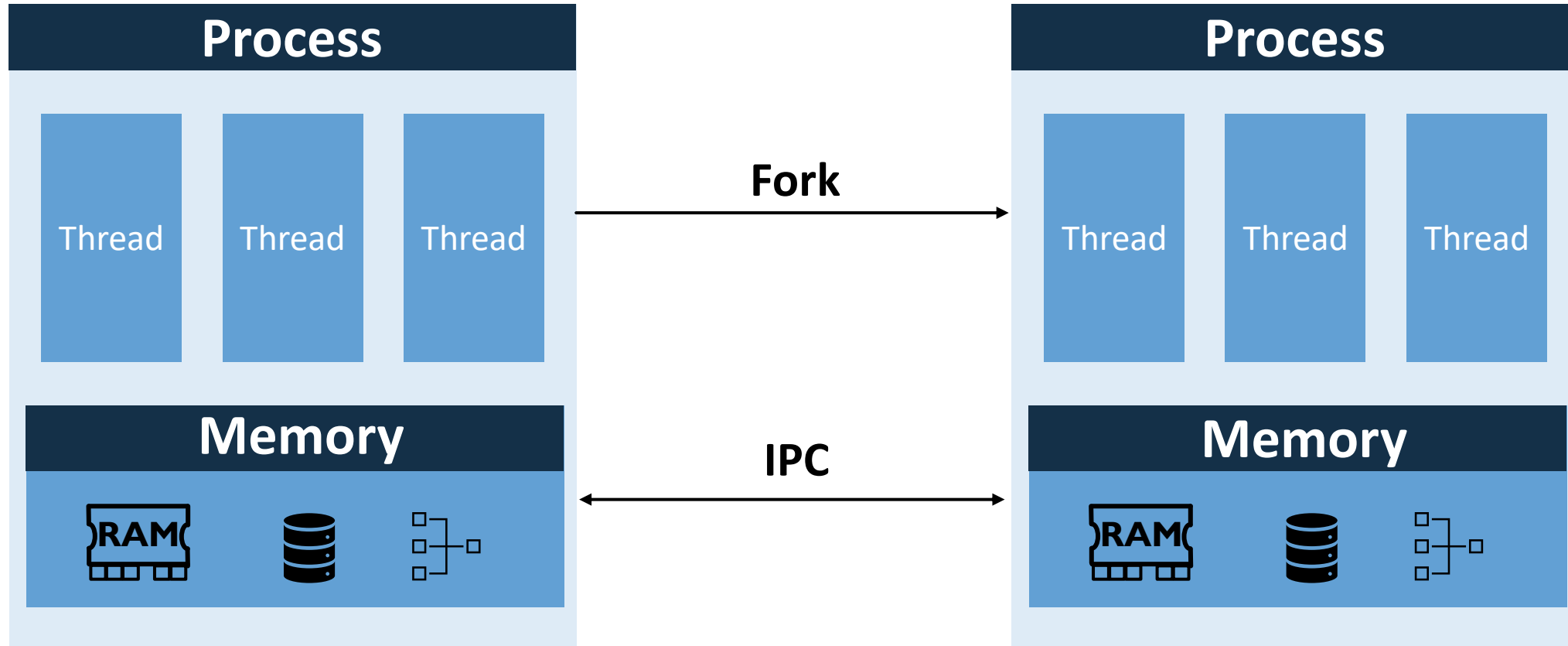
POSIX Process Model



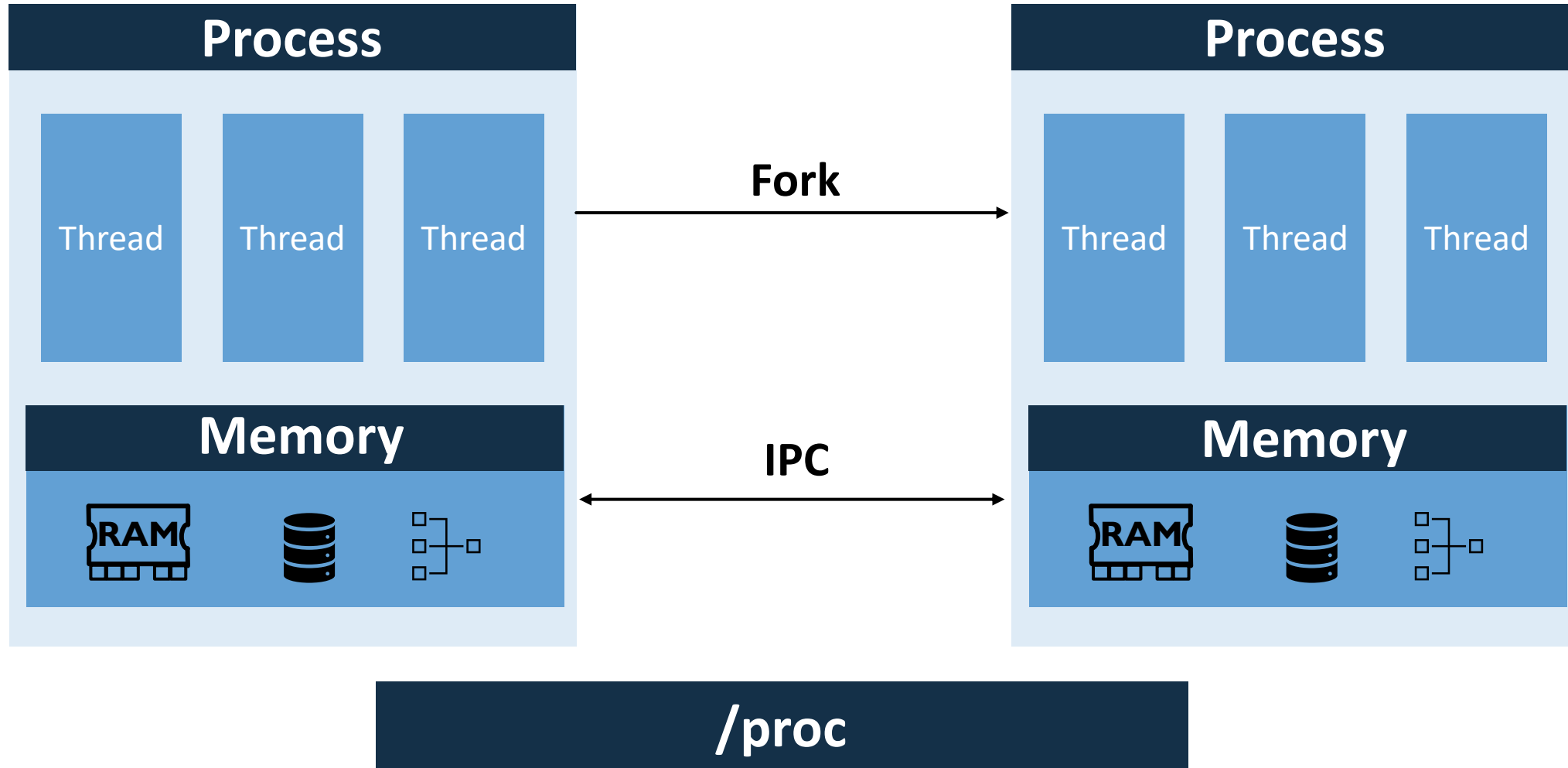
POSIX Process Model



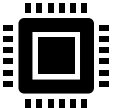




POSIX Process Model



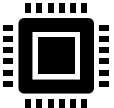




POSIX Process Model



Cloud Process

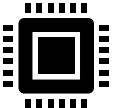




	IaaS	CaaS	FaaS
	Persistent	Persistent	Ephemeral
	Provisioned	Provisioned	Pay-as-you-go
	Persistent, local	Persistent, local	Ephemeral, remote
	Months	Hours, days	Minutes, seconds
	Manual, minutes	Semi-automatic, minutes & seconds	Automatic, seconds

Cloud Process

	IaaS	CaaS	PaaS	FaaS
	Persistent	Persistent	Ephemeral	Ephemeral
	Provisioned	Provisioned	Pay-as-you-go	Pay-as-you-go
	Persistent, local	Persistent, local	Persistent, local	Ephemeral, remote
	Months	Hours, days	Minutes, hours	Minutes, seconds
	Manual, minutes	Semi-automatic, minutes & seconds	Automatic, seconds	Automatic, seconds

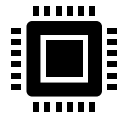




Cloud Process

Durable state of a container

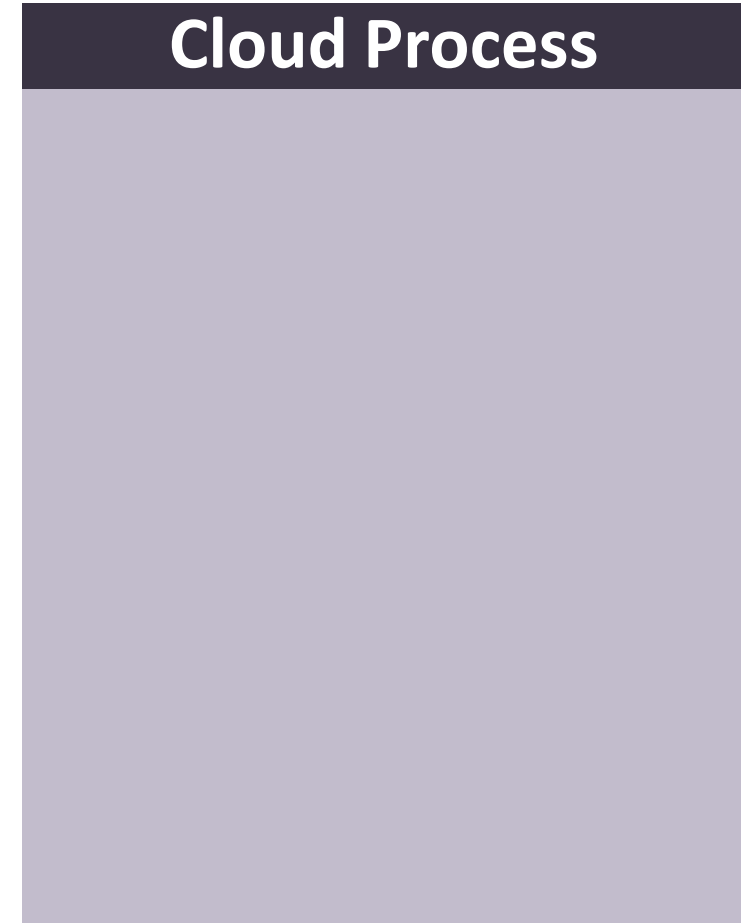
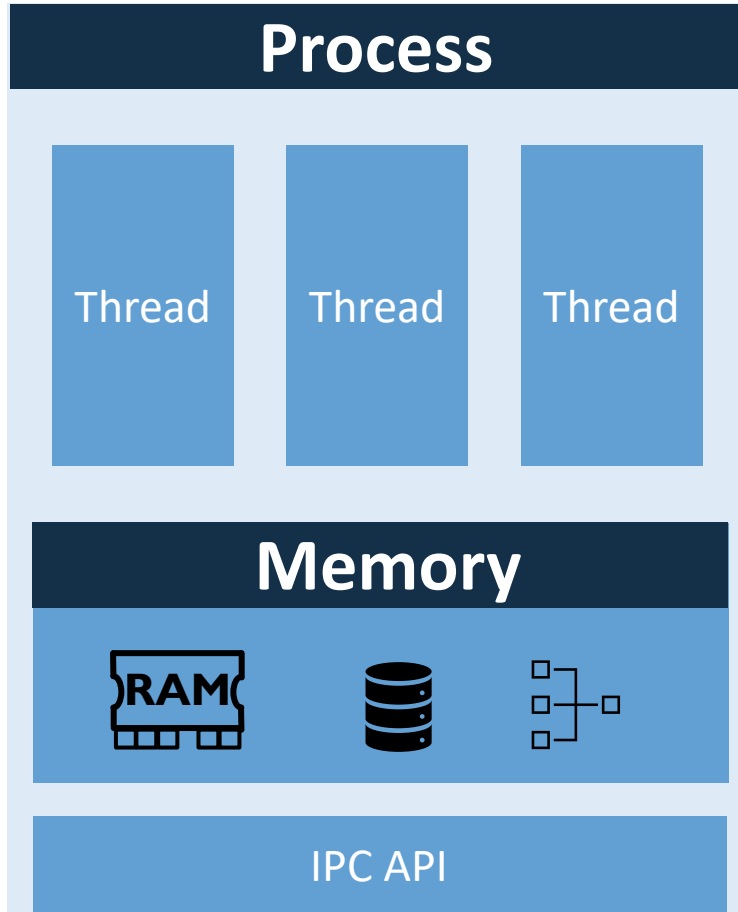
	IaaS	CaaS	PaaS	FaaS
	Persistent	Persistent	Ephemeral	Ephemeral
	Provisioned	Provisioned	Pay-as-you-go	Pay-as-you-go
	Persistent, local	Persistent, local	Persistent, local	Ephemeral, remote
	Months	Hours, days	Minutes, hours	Minutes, seconds
	Manual, minutes	Semi-automatic, minutes & seconds	Automatic, seconds	Automatic, seconds

Cloud Process

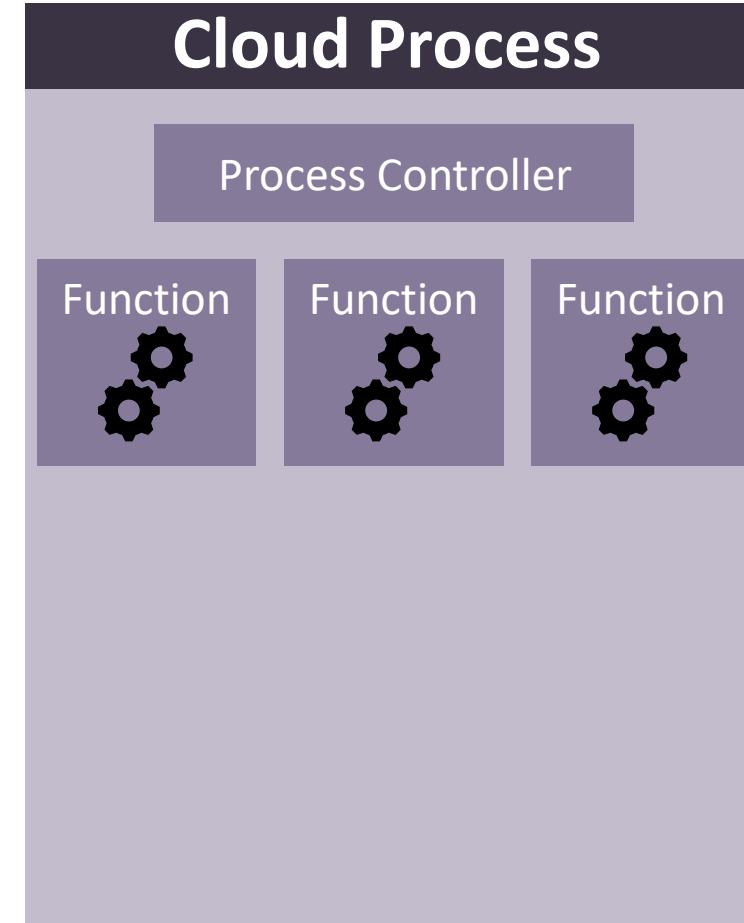
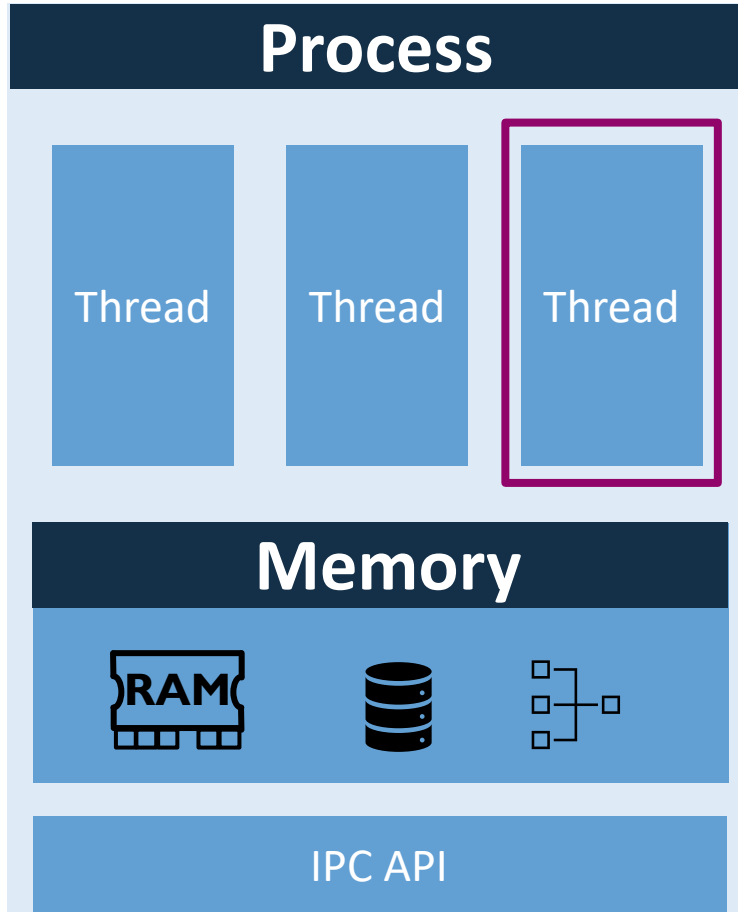
Durable state of a container and ephemeral compute of serverless.

	IaaS	CaaS	PaaS	FaaS
	Persistent	Persistent	Ephemeral	Ephemeral
	Provisioned	Provisioned	Pay-as-you-go	Pay-as-you-go
	Persistent, local	Persistent, local	Persistent, local	Ephemeral, remote
	Months	Hours, days	Minutes, hours	Minutes, seconds
	Manual, minutes	Semi-automatic, minutes & seconds	Automatic, seconds	Automatic, seconds

Cloud Process Model

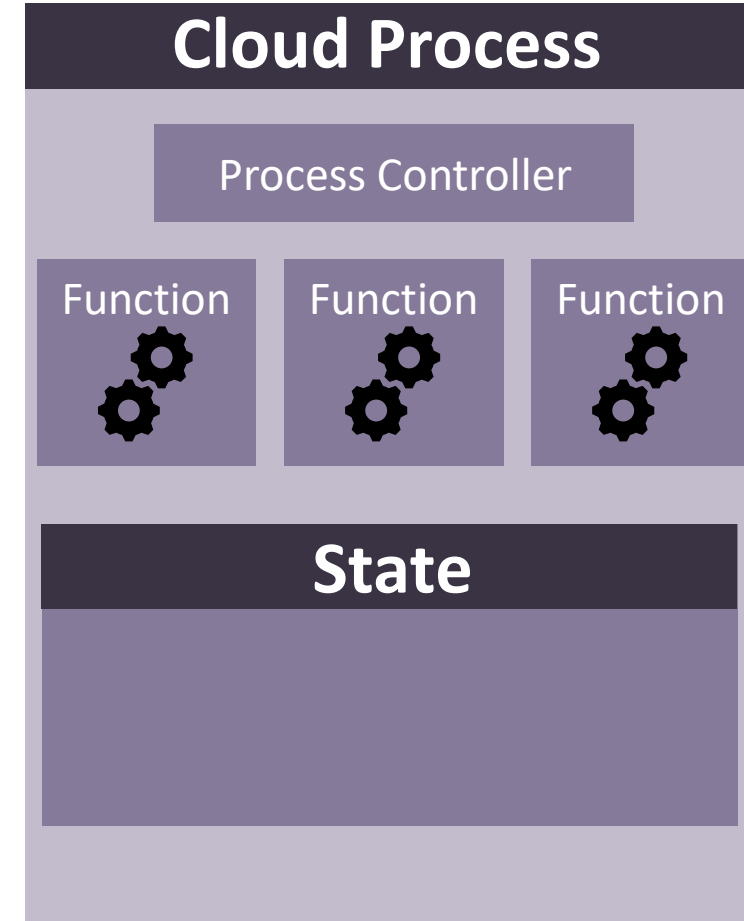
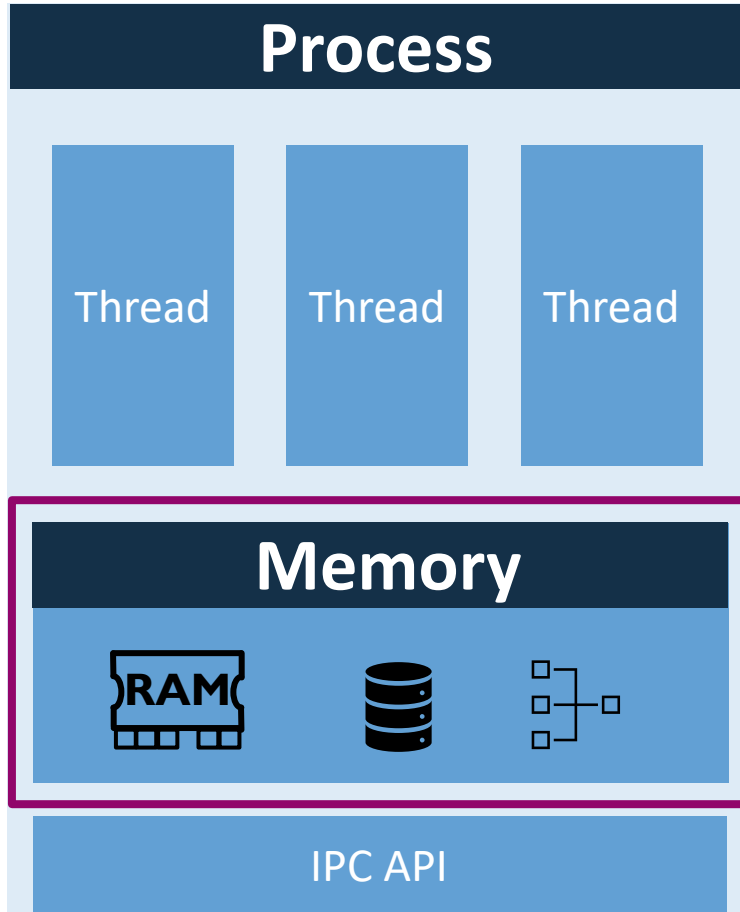


Cloud Process Model



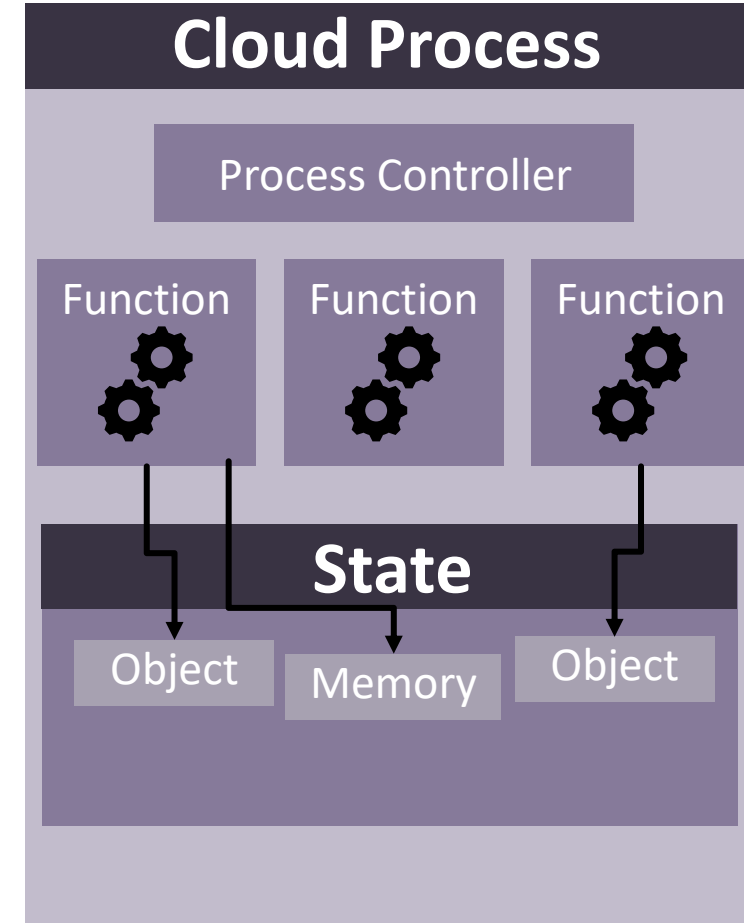
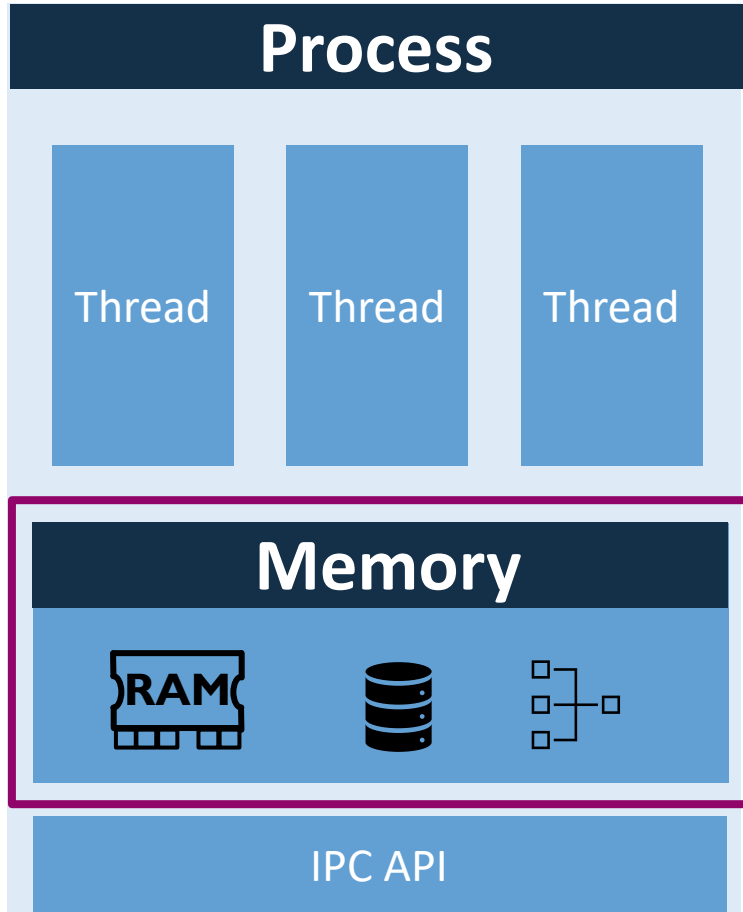
Functions represent computational threads within process.

Cloud Process Model



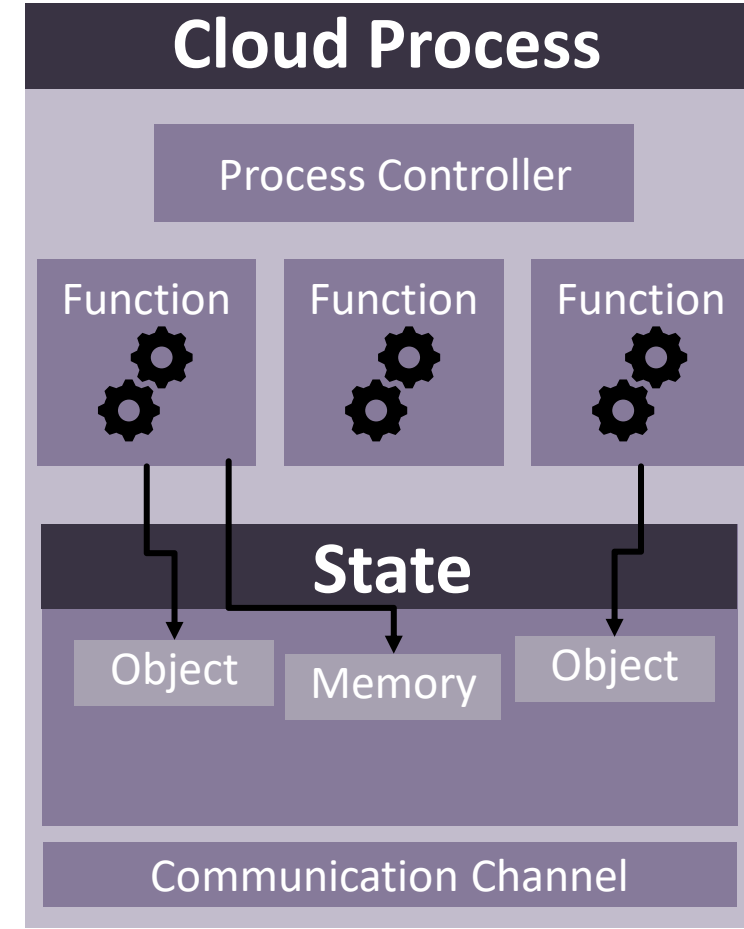
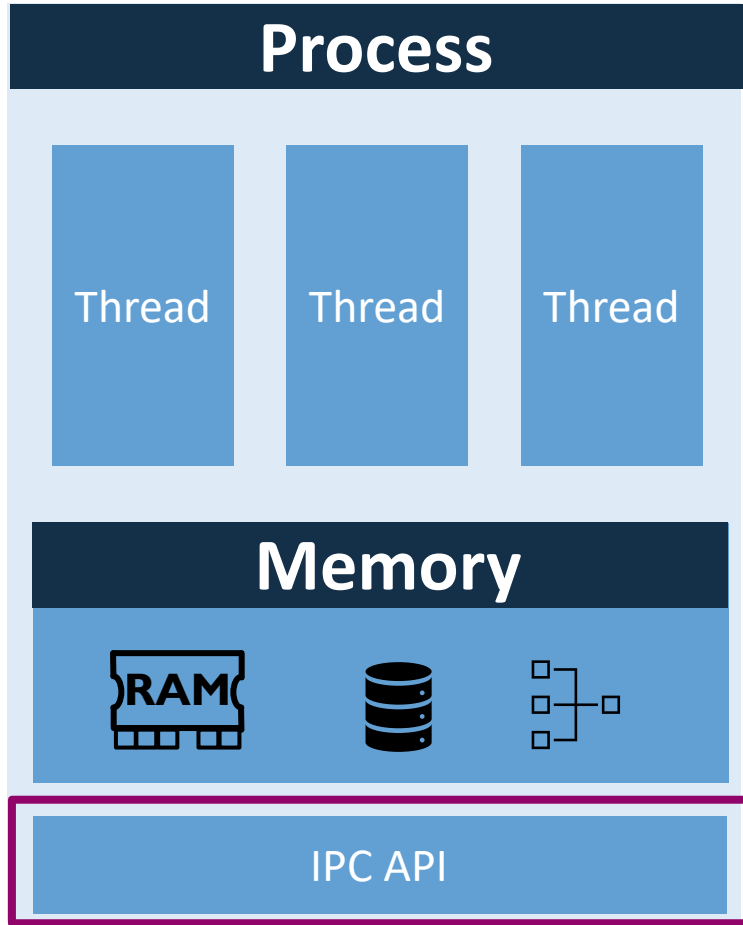
State is represented as a collection of generic objects

Cloud Process Model



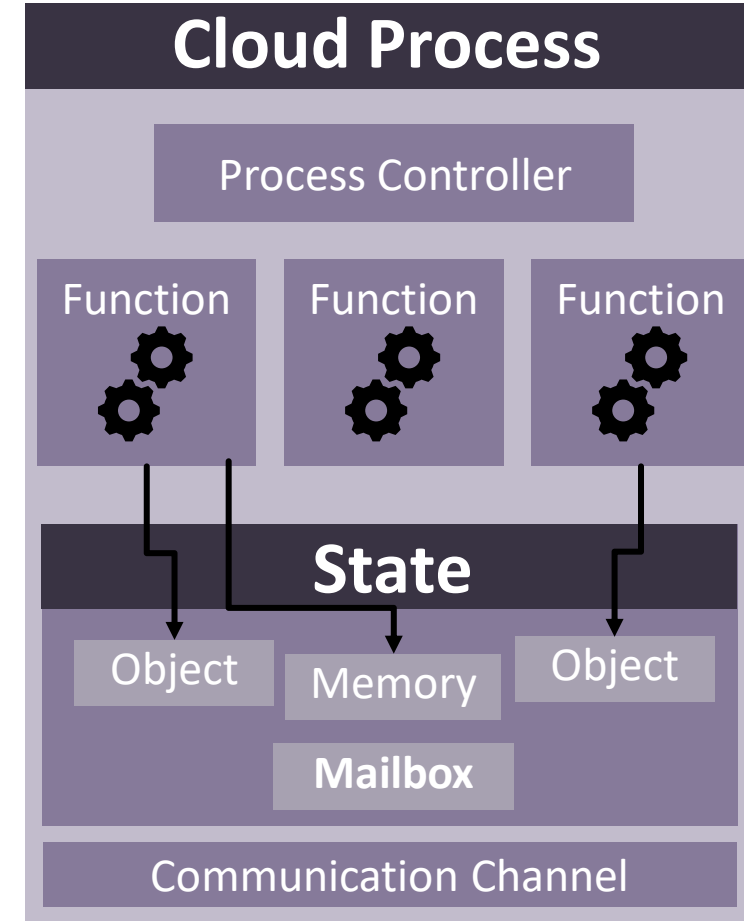
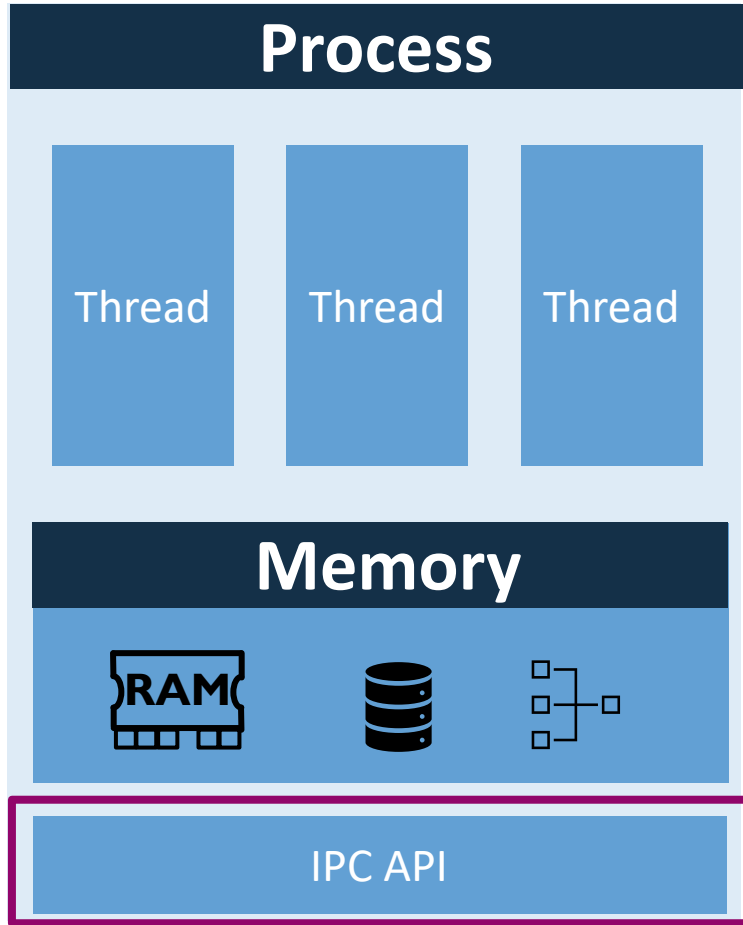
State is represented as a collection of generic objects

Cloud Process Model



Channels are an abstraction for actual data movement.

Cloud Process Model

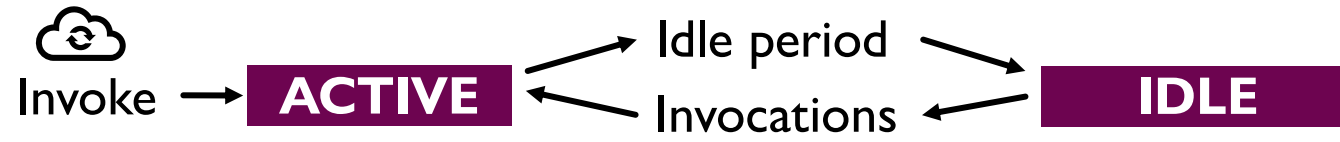


Channels are an abstraction for actual data movement.

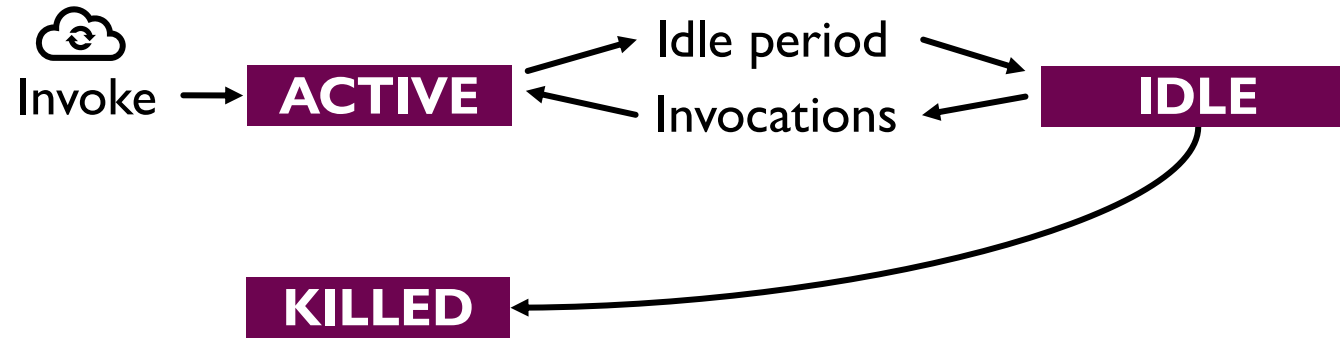
Serverless Function Lifetime



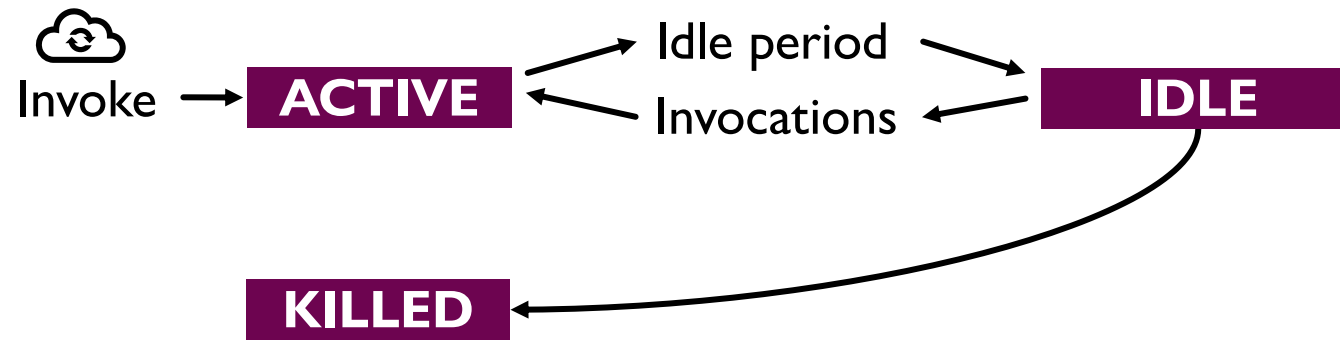
Serverless Function Lifetime



Serverless Function Lifetime

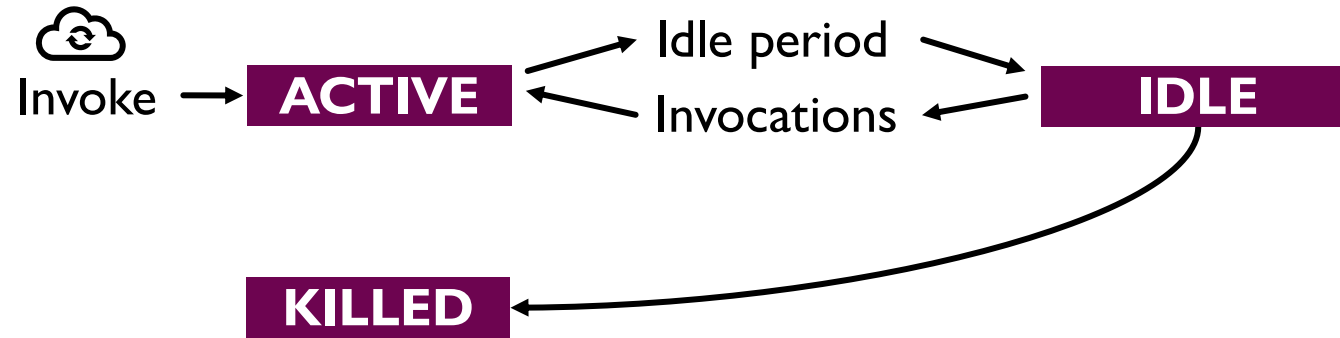




Serverless Function Lifetime



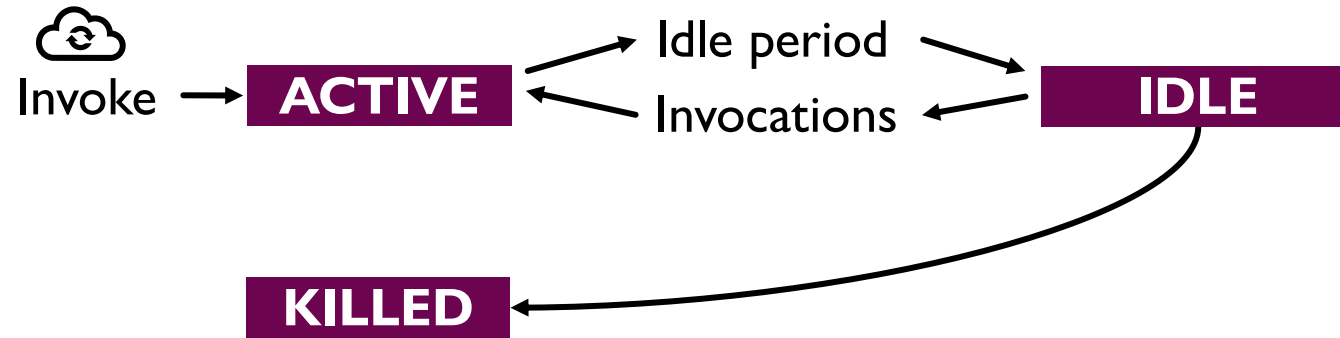
- 
 Users do not control function allocation and status.

Serverless Function Lifetime



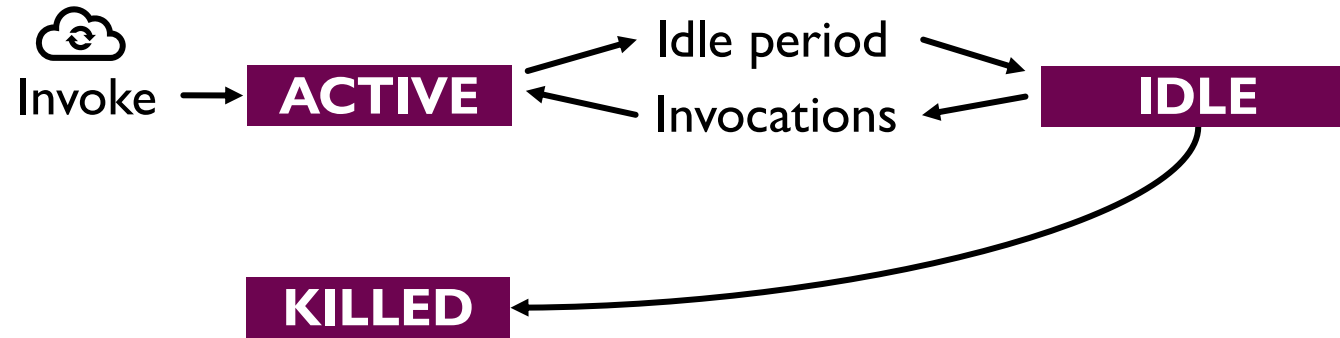
- 
 Users do not control function allocation and status.
- 
 Idle instances are stopped automatically.

Serverless Function Lifetime



OS Process states: ready, running, terminated, waiting.

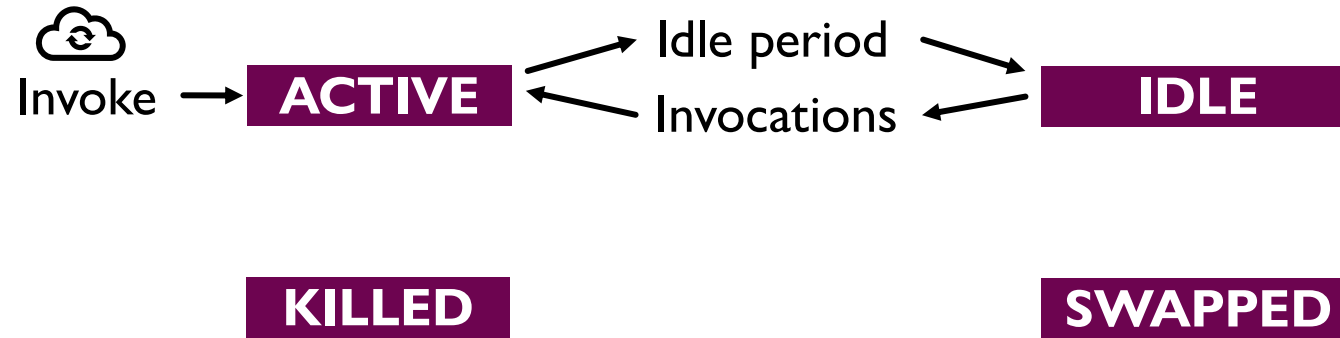
Serverless Function Lifetime



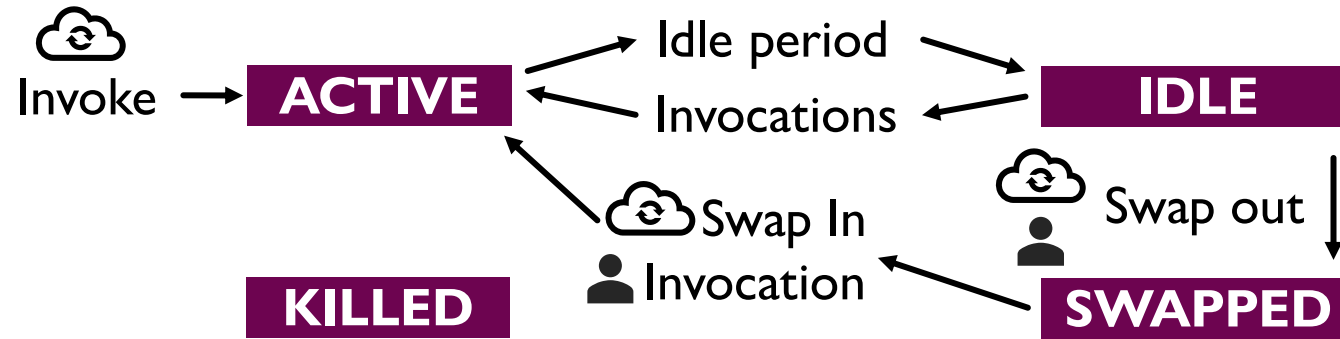
OS Process states: ready, running, terminated, waiting.

Cloud Process states: idle, active, killed, **swapped out**.

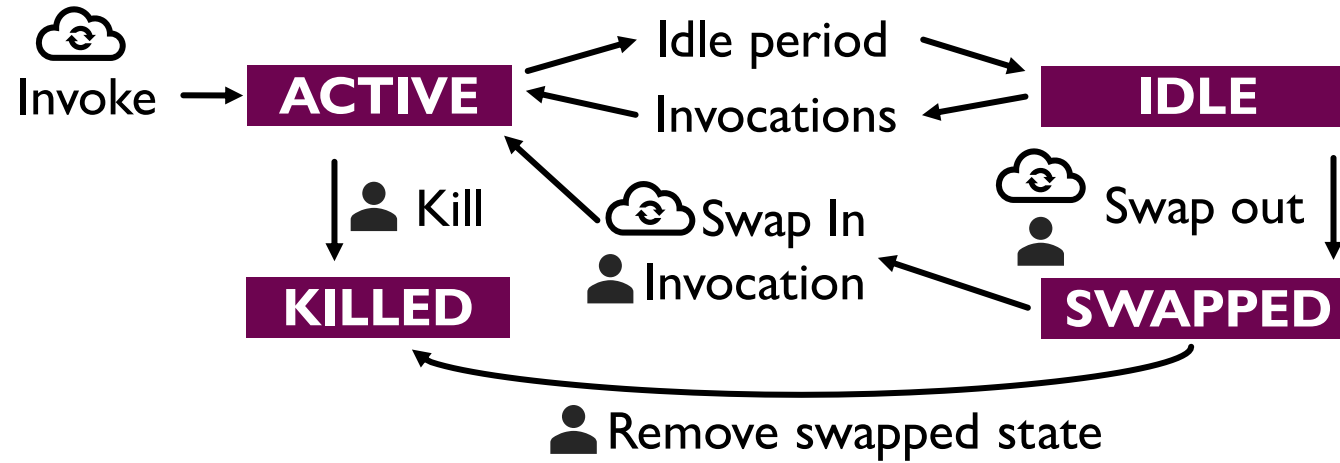
Serverless Process Lifetime



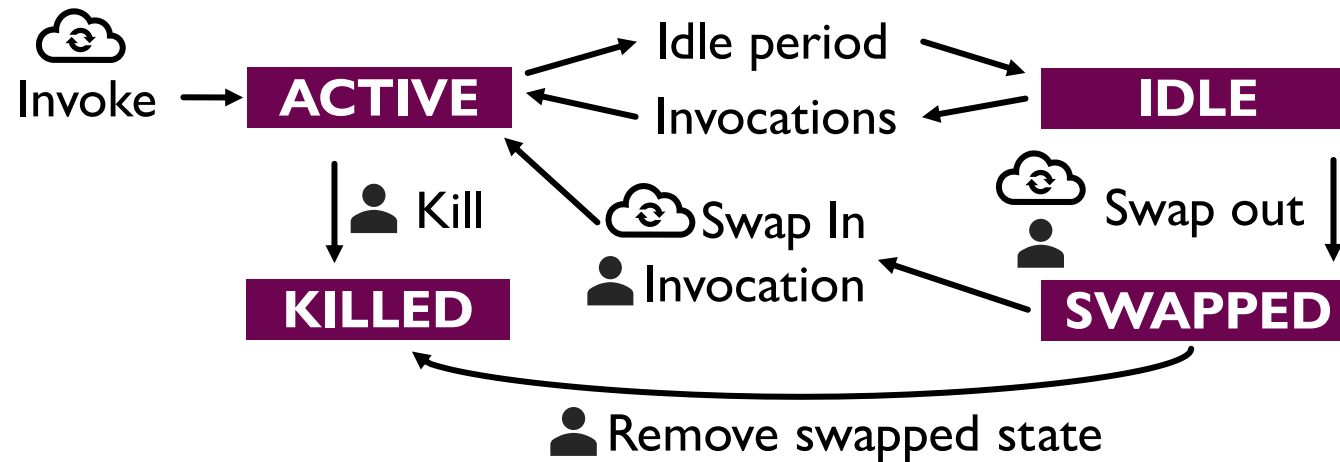
Serverless Process Lifetime



Serverless Process Lifetime

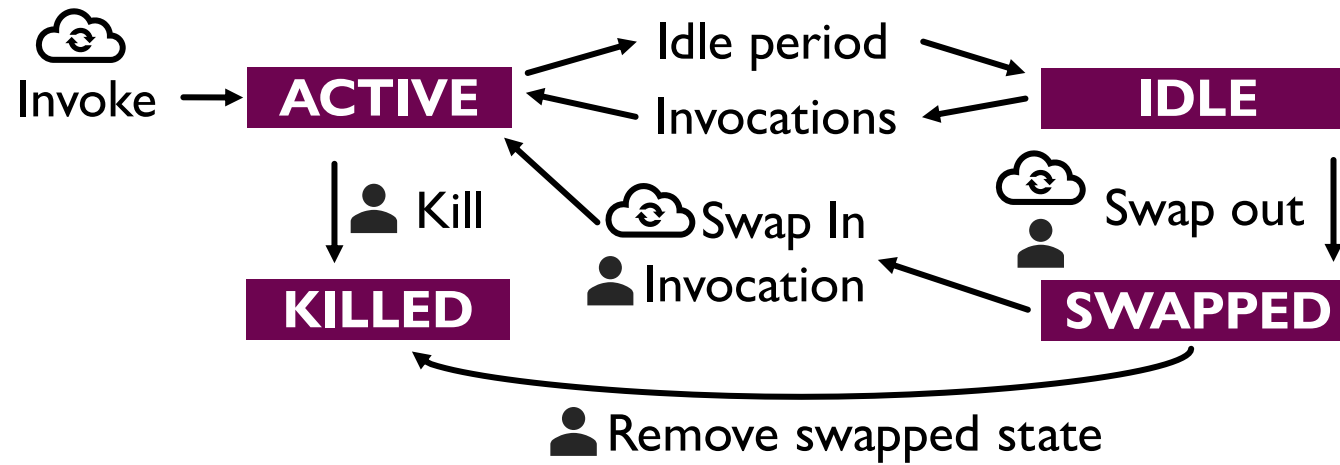





Serverless Process Lifetime



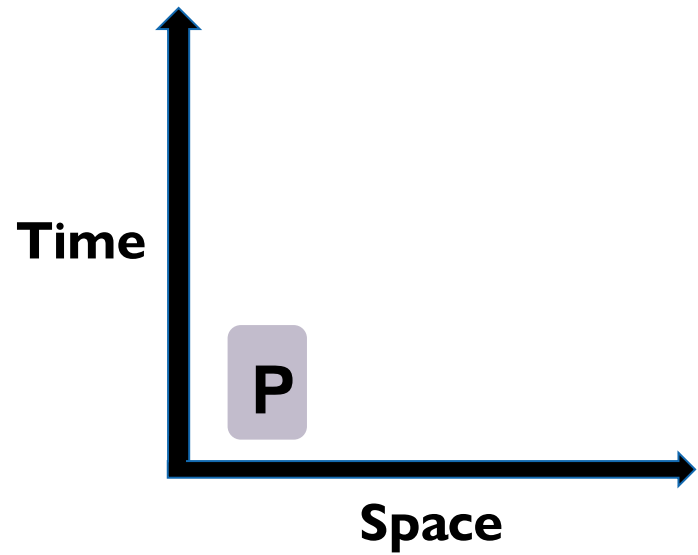
- ✓ Users do not control function allocation and status.
- ✓ Idle instances are stopped automatically.

Serverless Process Lifetime

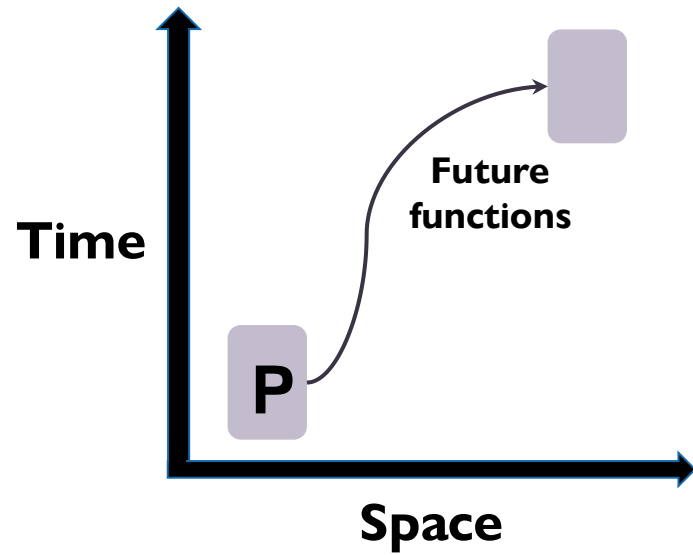


- 
 Users do not control function allocation and status.
- 
 Idle instances are stopped automatically.
- 
 Invocations with the same PID access the same state.

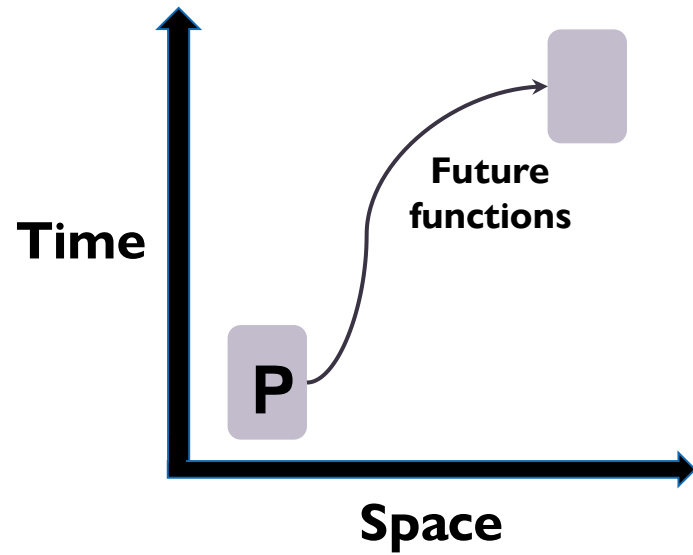
Inter-Process Communication



Inter-Process Communication

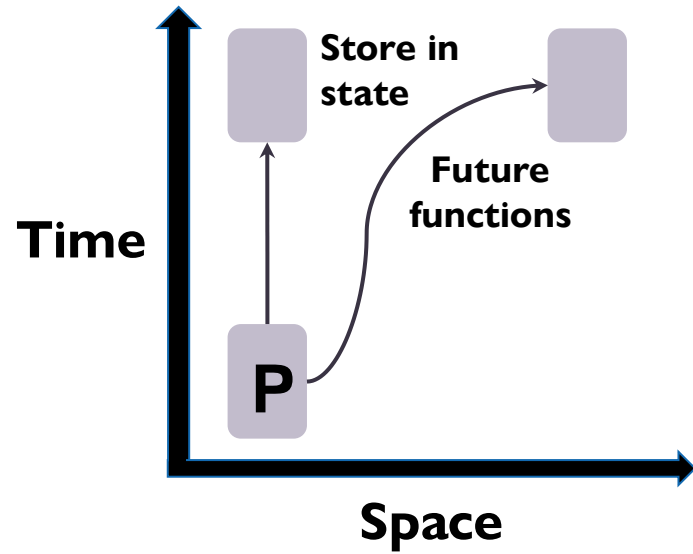


Inter-Process Communication



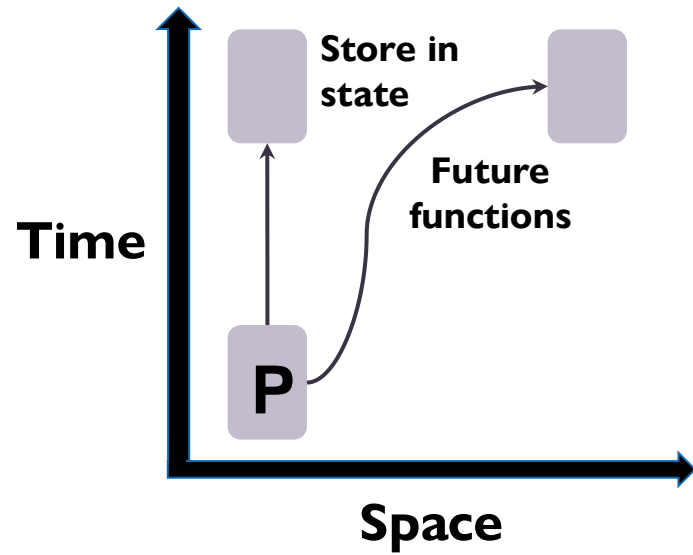
```
invoke(pid_destination, function, data)
```

Inter-Process Communication



```
invoke(pid_destination, function, data)
```

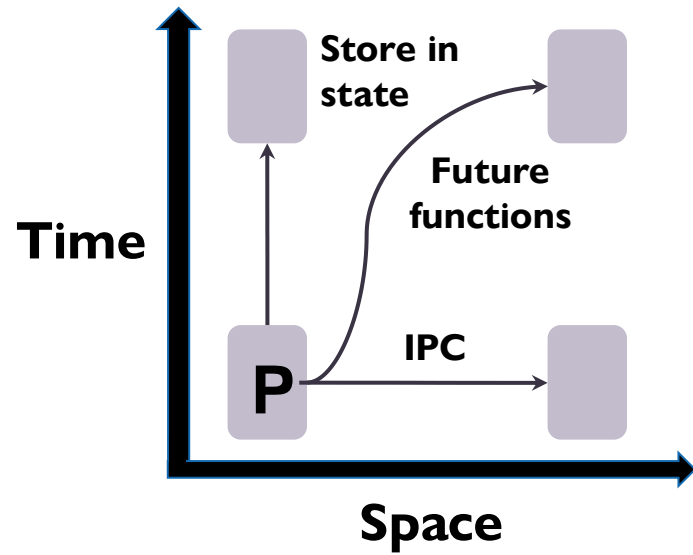
Inter-Process Communication



```
invoke(pid_destination, function, data)
```

```
state(key, data)
data = state(key)
```

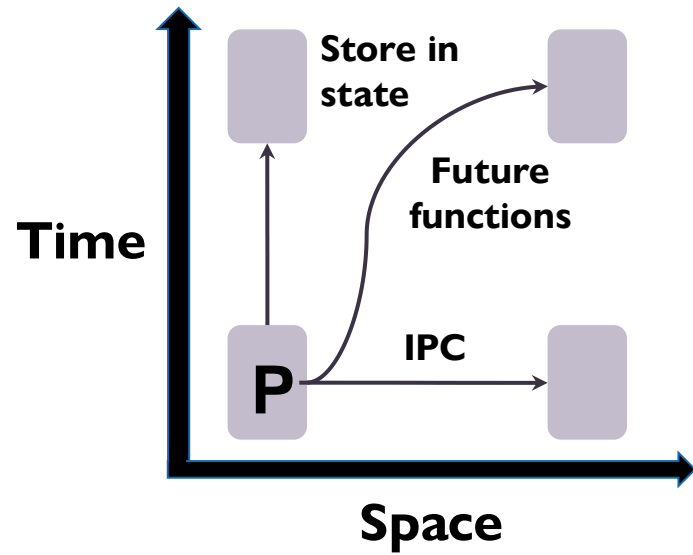
Inter-Process Communication



```
invoke(pid_destination, function, data)
```

```
state(key, data)
data = state(key)
```

Inter-Process Communication



```
invoke(pid_destination, function, data)
```

```
state(key, data)
data = state(key)
```

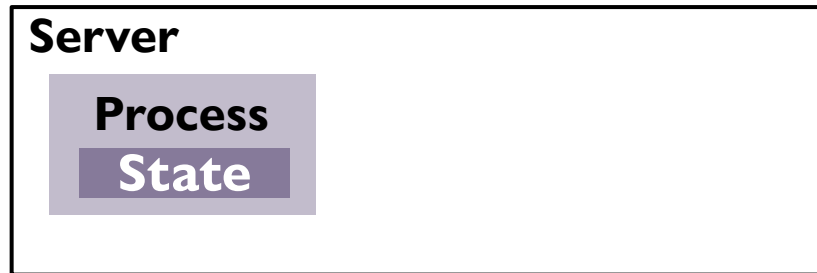
```
send(pid_destination, key, data)
data = recv(pid_source, key)
```

PraaS: Process-as-a-Service

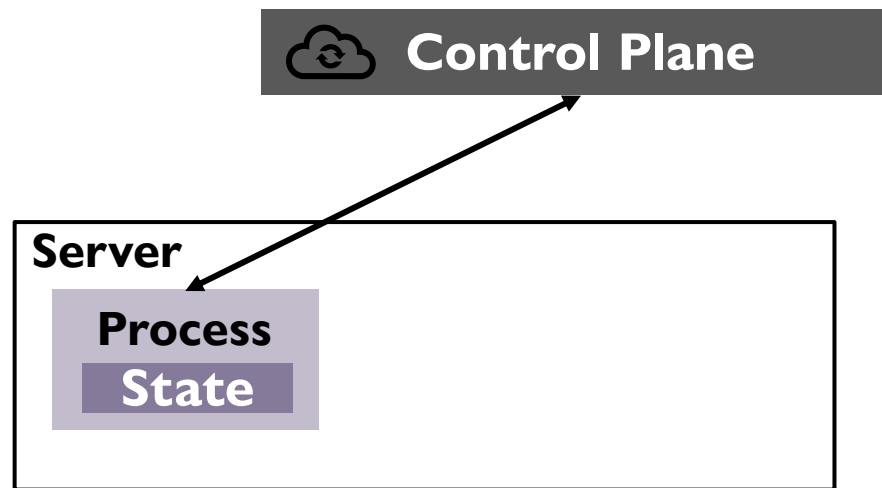
PraaS: Process-as-a-Service

Process
State

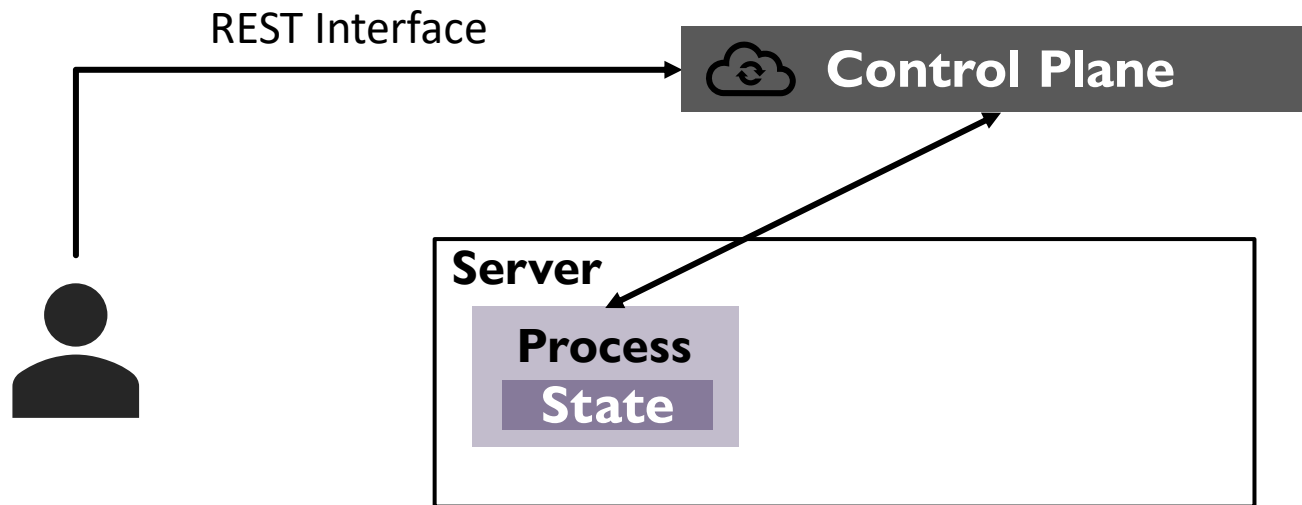
PraaS: Process-as-a-Service



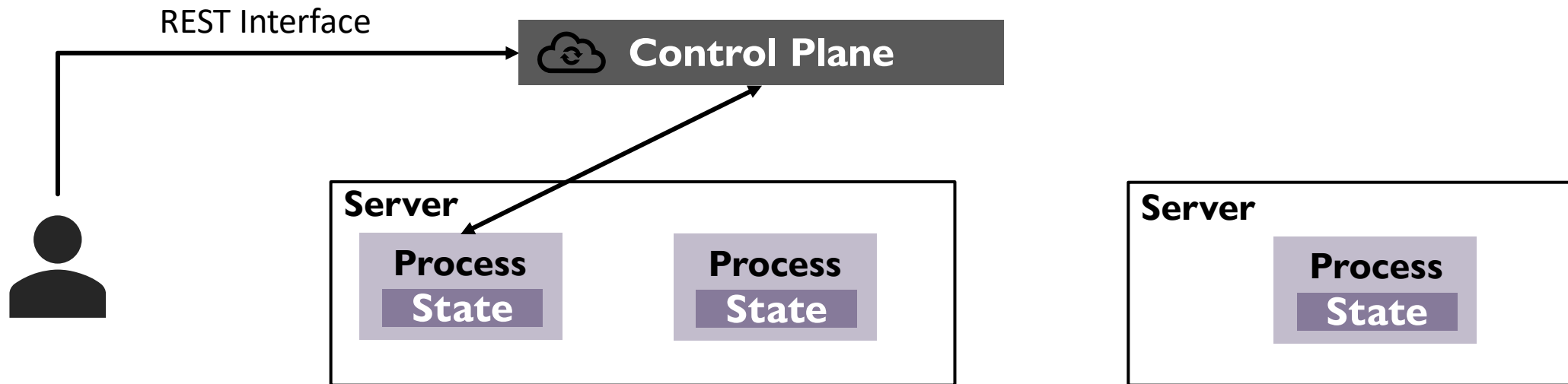
PraaS: Process-as-a-Service



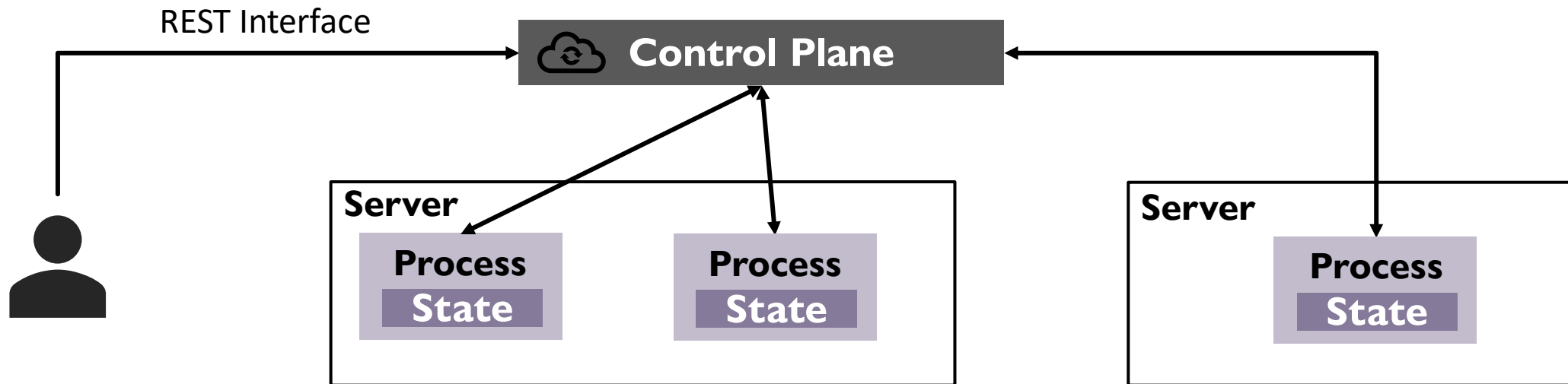
PraaS: Process-as-a-Service



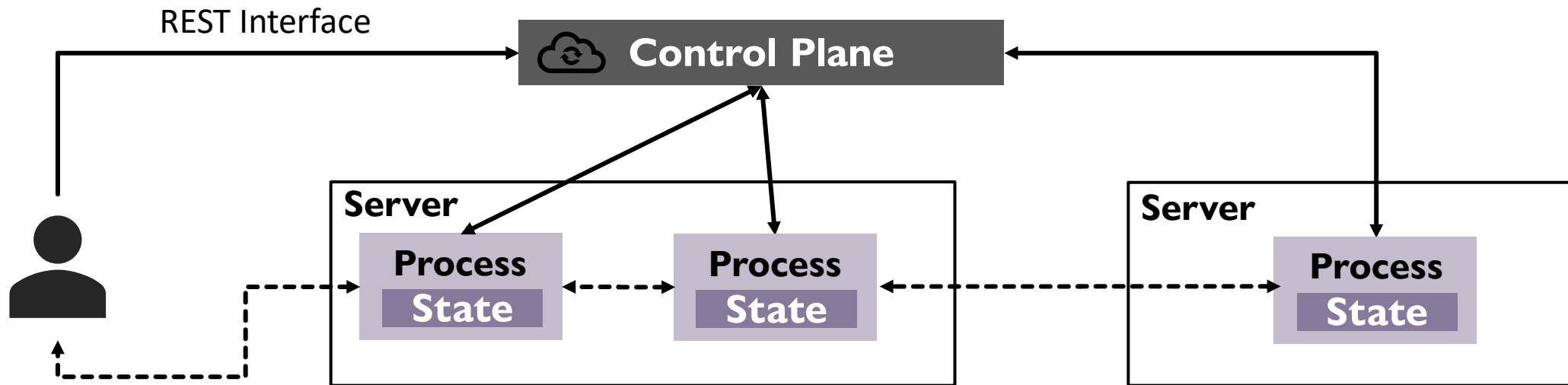
PraaS: Process-as-a-Service



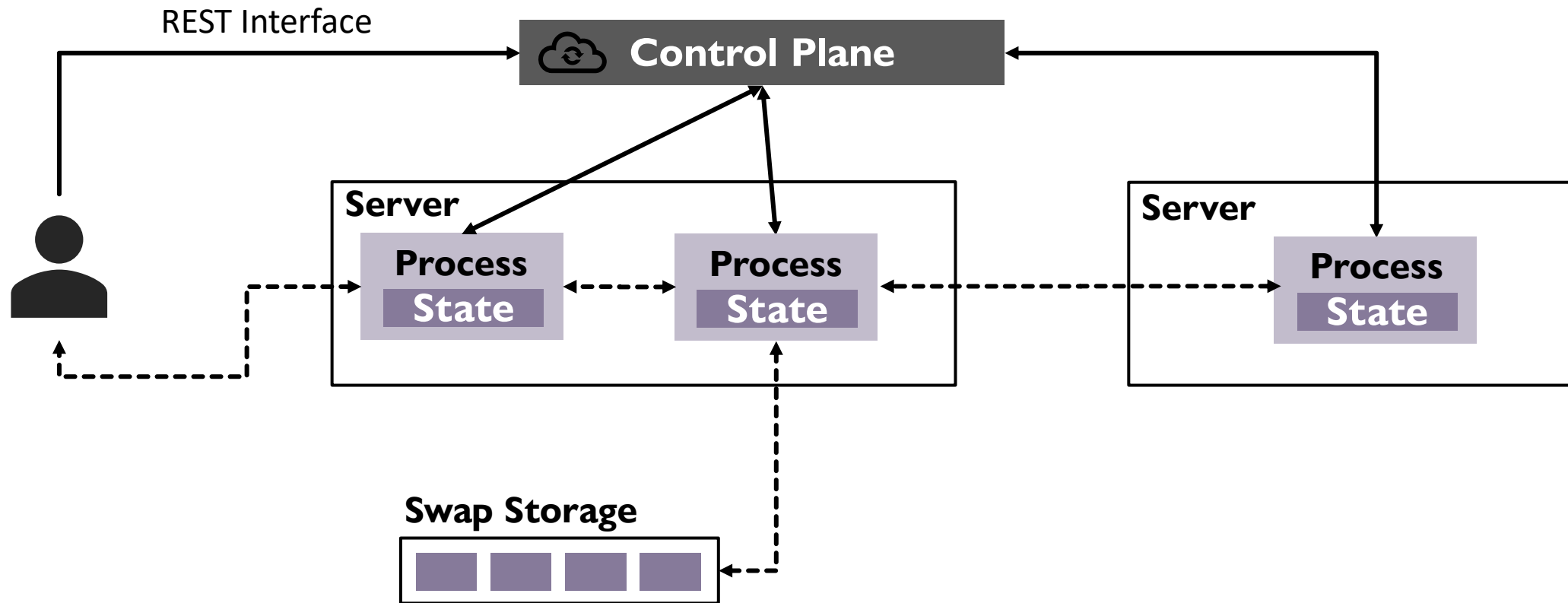
PraaS: Process-as-a-Service



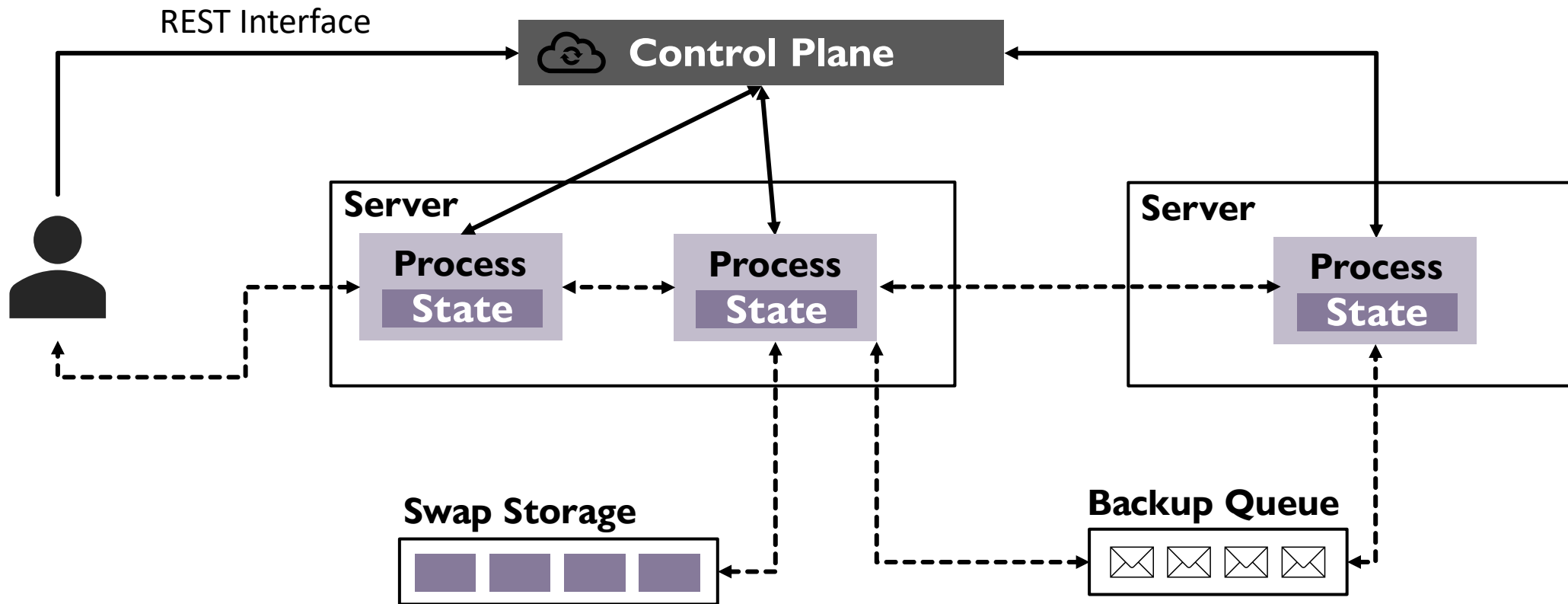
PraaS: Process-as-a-Service



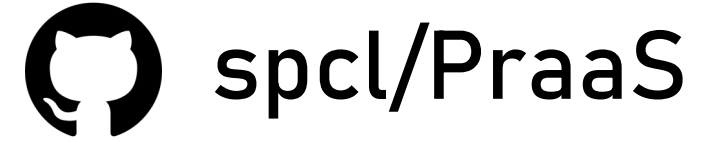
PraaS: Process-as-a-Service



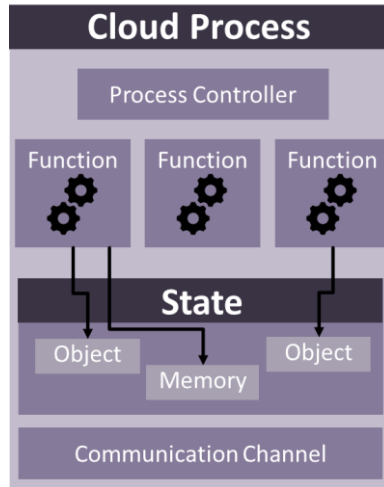
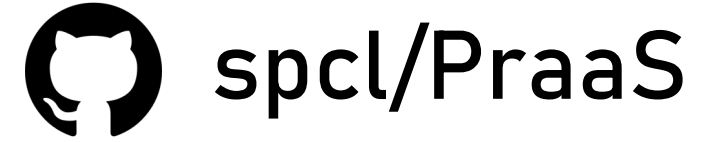
PraaS: Process-as-a-Service



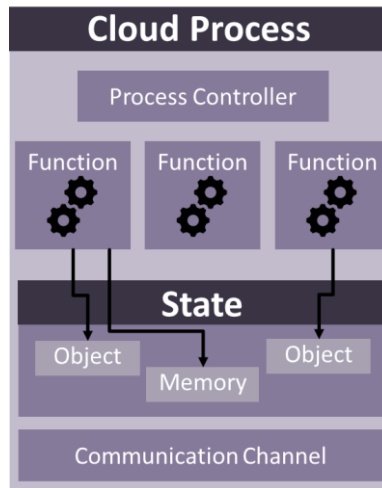
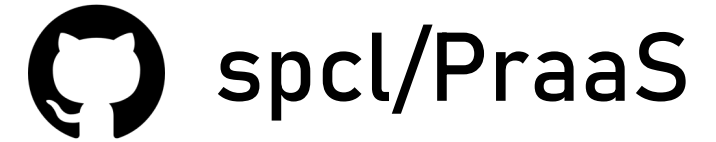
Prototype Implementation



Prototype Implementation

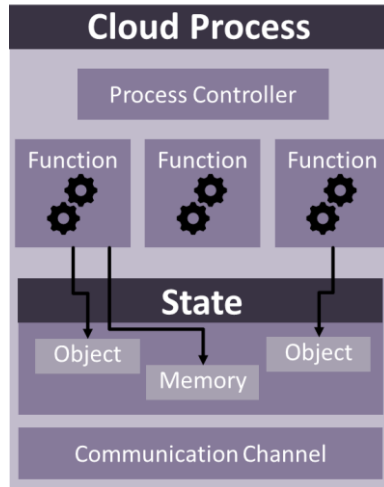


Prototype Implementation

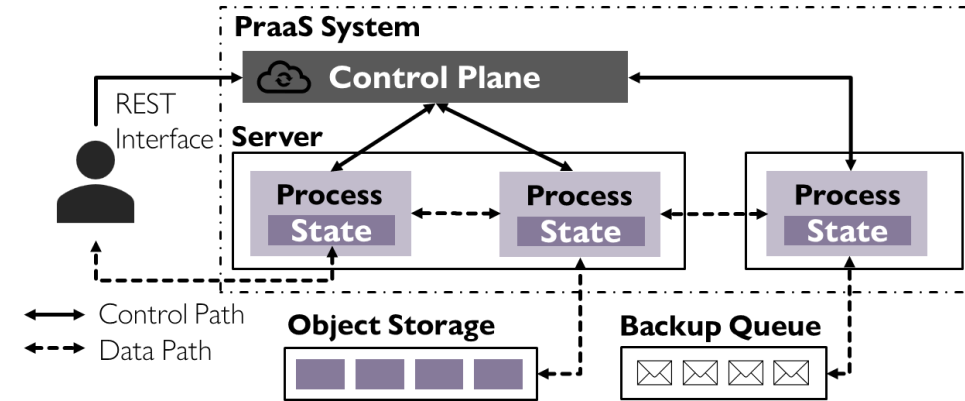
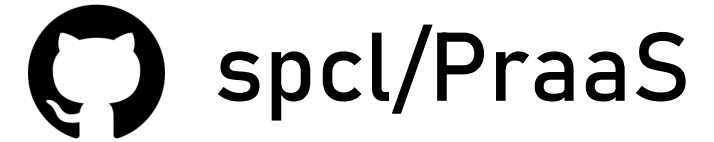


- Multi-process solution for functions.
- Communication over POSIX queues.
- Inter-process communication over TCP.

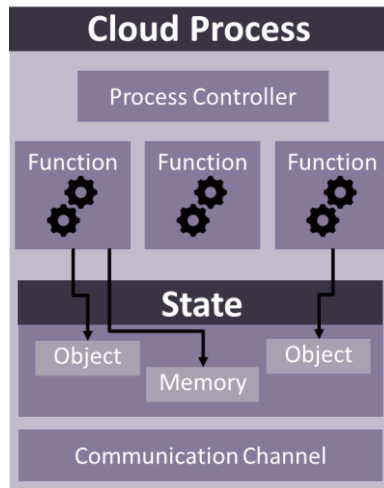
Prototype Implementation



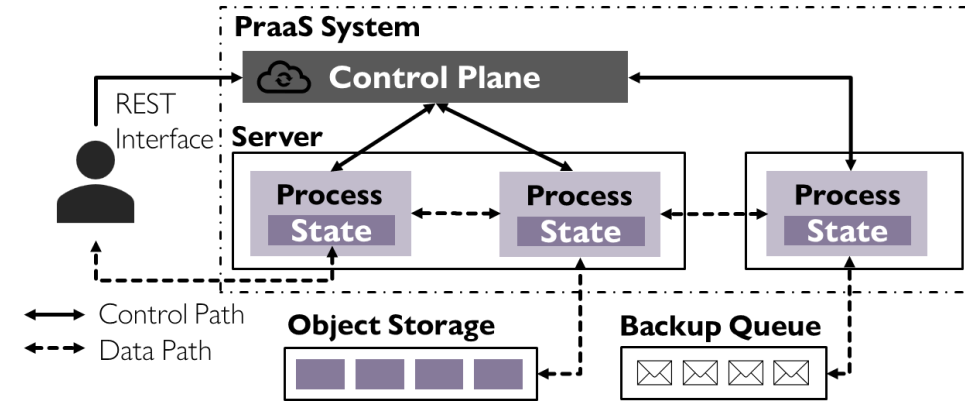
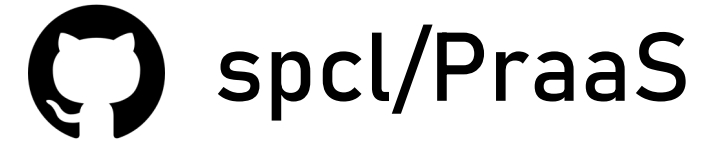
- Multi-process solution for functions.
- Communication over POSIX queues.
- Inter-process communication over TCP.



Prototype Implementation

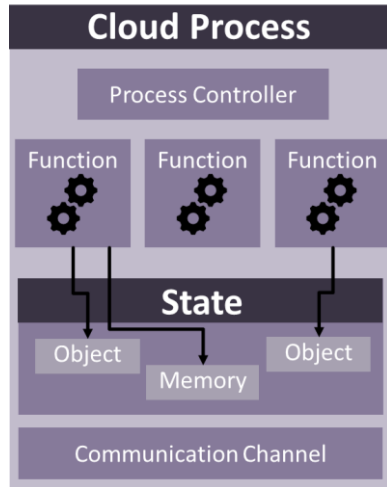


- Multi-process solution for functions.
- Communication over POSIX queues.
- Inter-process communication over TCP.

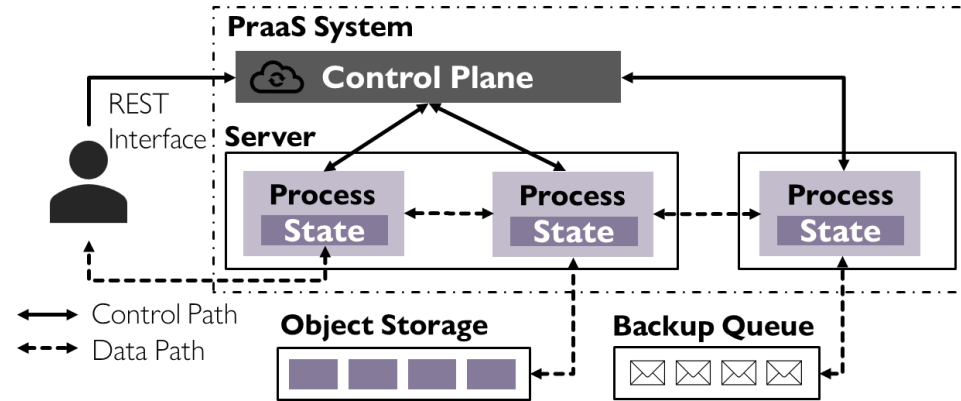
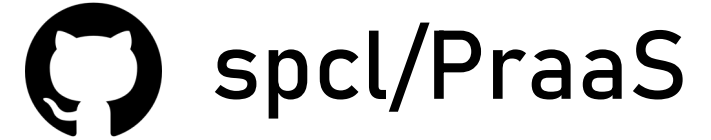


- Custom control plane.
- Allocating processes on AWS Fargate and Docker containers.

Prototype Implementation



- Multi-process solution for functions.
- Communication over POSIX queues.
- Inter-process communication over TCP.

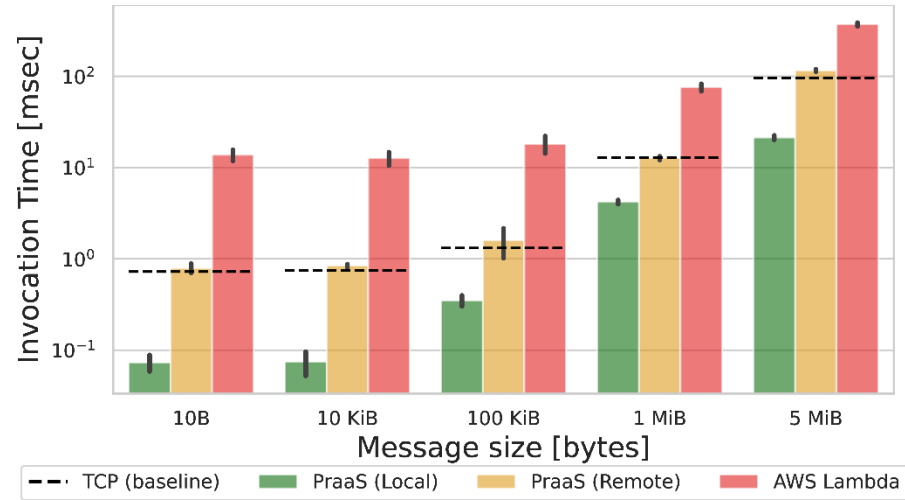


- Custom control plane.
- Allocating processes on AWS Fargate and Docker containers.
- Modified scheduler for Knative and Kubernetes to support swapping.

Microbenchmarks Evaluation

Microbenchmarks Evaluation

Data Plane Invocations

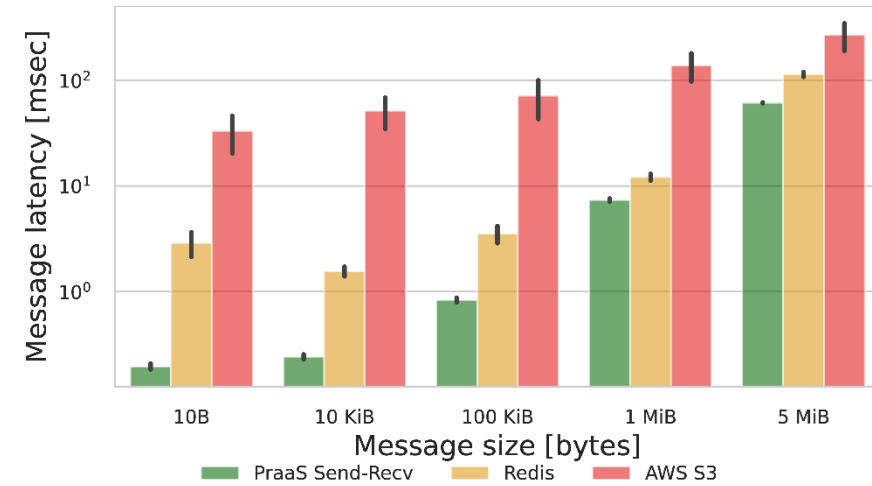


Microbenchmarks Evaluation

Data Plane Invocations

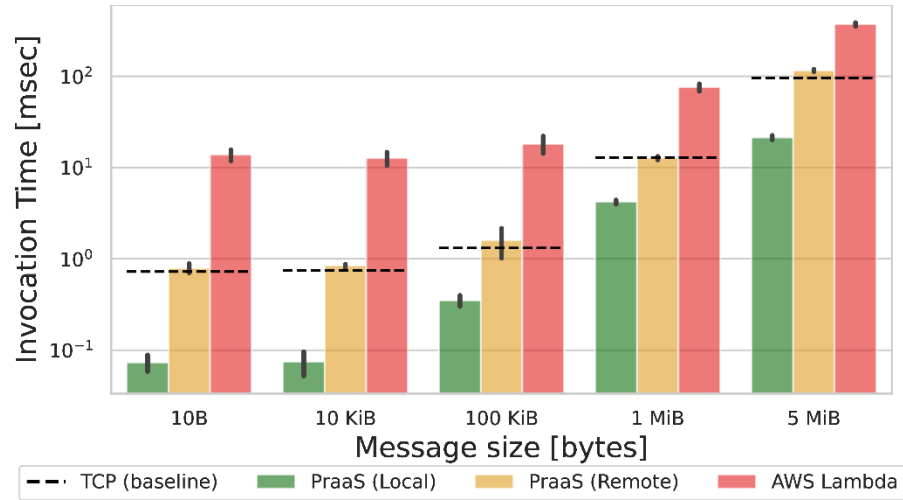


Inter-Process Communication

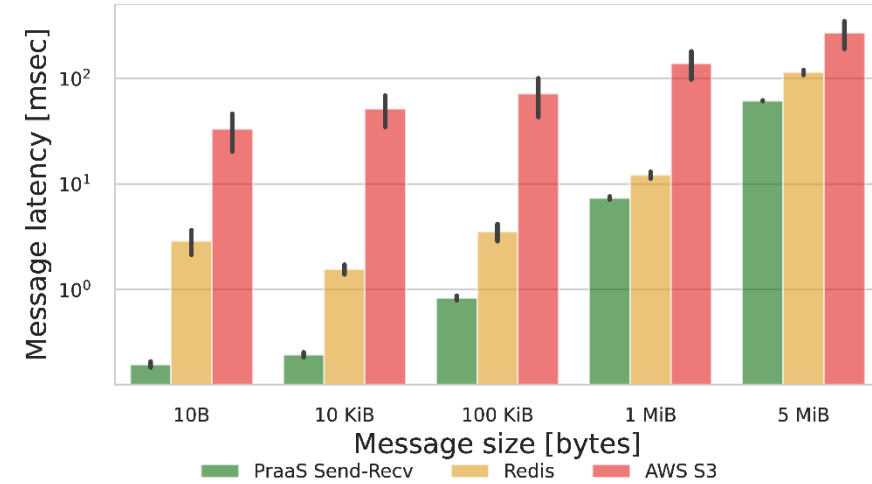


Microbenchmarks Evaluation

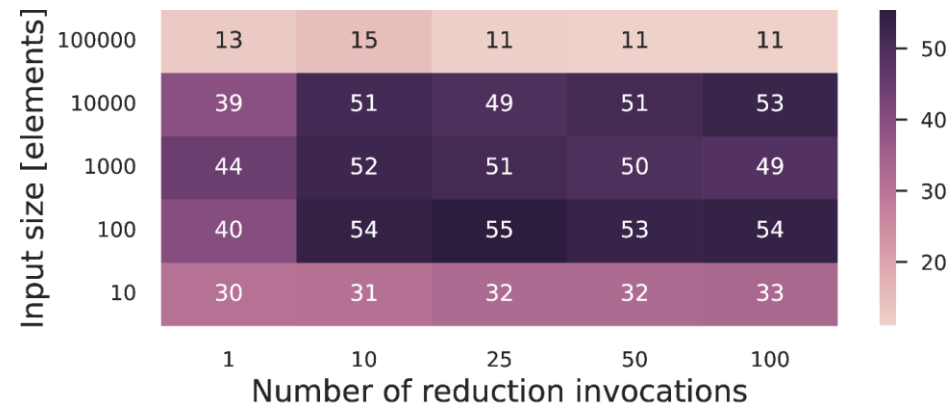
Data Plane Invocations










Inter-Process Communication



State Access

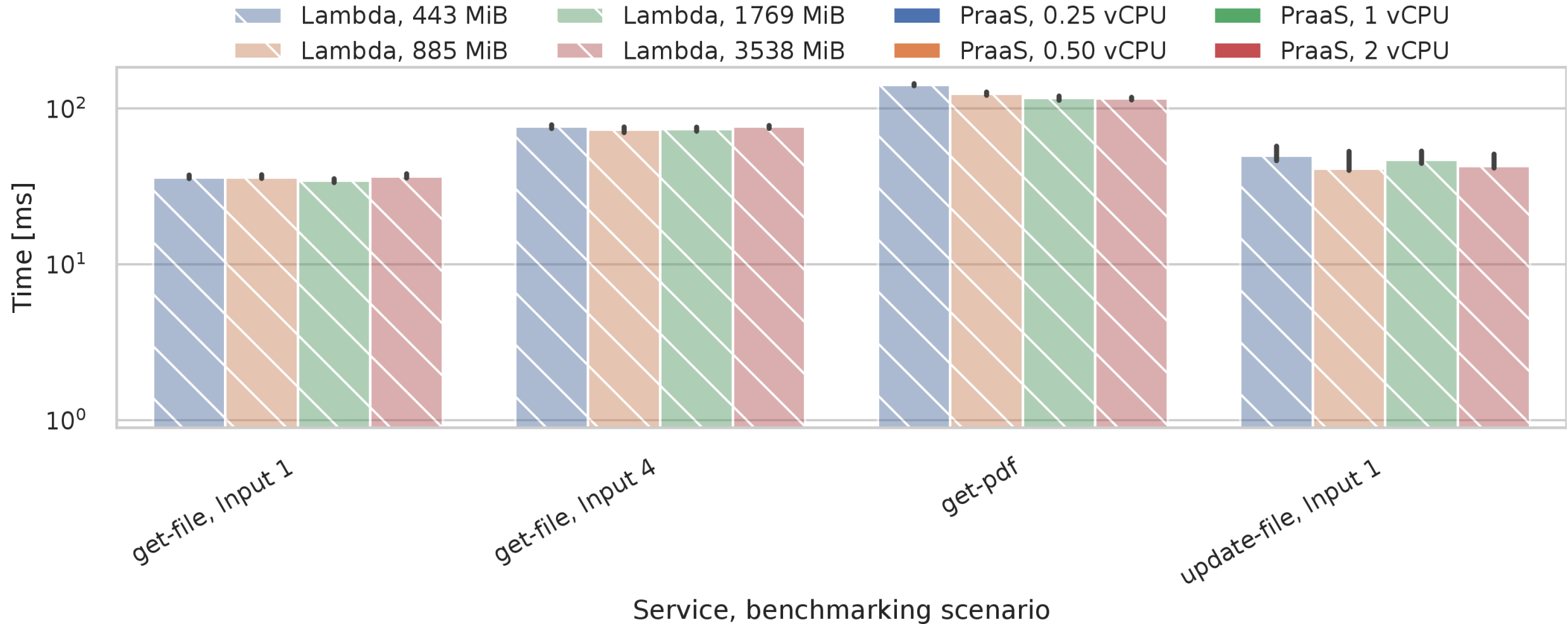


Benchmark: LaTeX Microservice

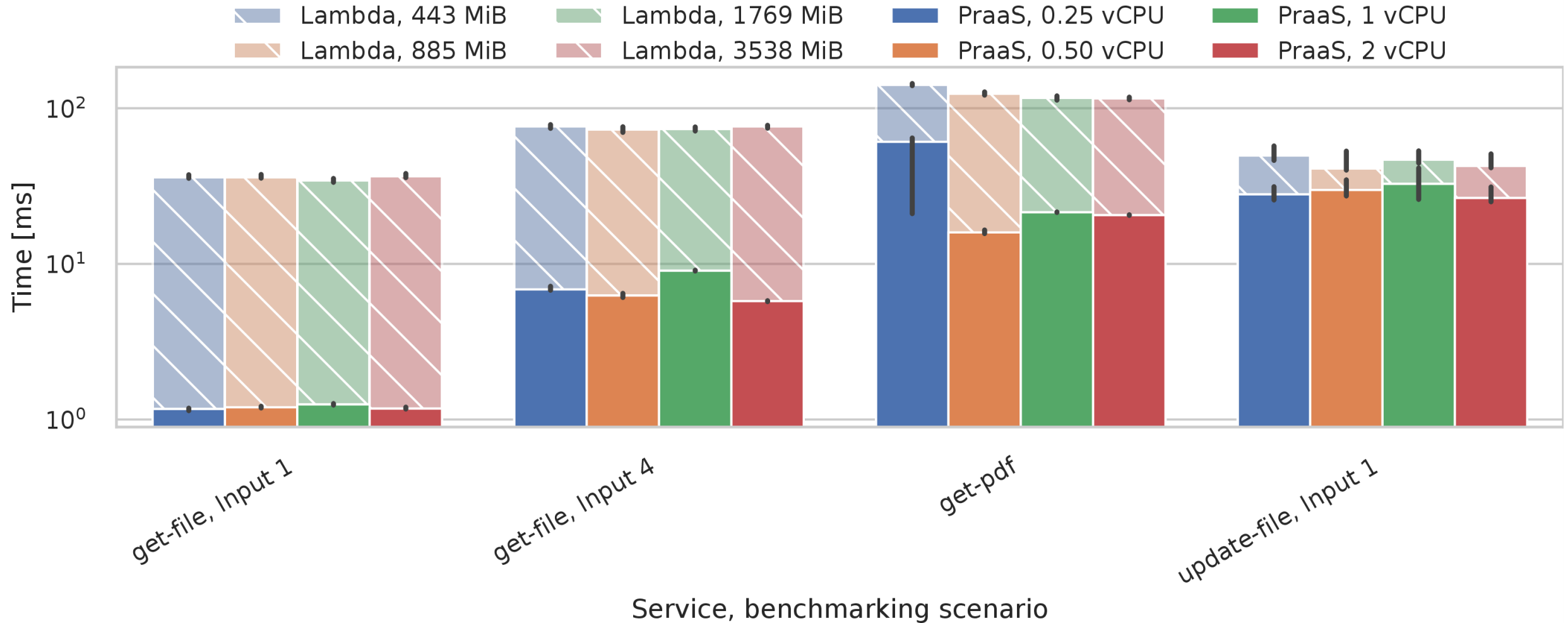
-  Lambda, 443 MiB
-  Lambda, 1769 MiB
-  PraaS, 0.25 vCPU
-  PraaS, 1 vCPU
-  Lambda, 885 MiB
-  Lambda, 3538 MiB
-  PraaS, 0.50 vCPU
-  PraaS, 2 vCPU



Benchmark: LaTeX Microservice



Benchmark: LaTeX Microservice



Serverless Solutions for HPC

Serverless Solutions for HPC



[spcl/serverless-benchmarks](https://github.com/spcl/serverless-benchmarks)

Serverless Solutions for HPC



[spcl/serverless-benchmarks](#)



[spcl/rFaaS](#)

Serverless Solutions for HPC



[spcl/serverless-benchmarks](#)



[spcl/rFaaS](#)



[spcl/fmi](#)

Serverless Solutions for HPC



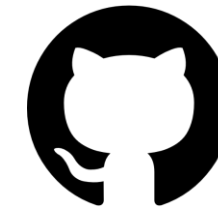
[spcl/serverless-benchmarks](#)



[spcl/rFaaS](#)



[spcl/fmi](#)



[spcl/FaaSKeeper](#)



Marcin Copik
ETH Zurich



Alexandru Calotoiu
ETH Zurich



Gyorgy Rethy
Oracle Labs



Roman Böhringer
OpenCore GmbH





Rodrigo Bruno
INESC-ID




Torsten Hoefler
ETH Zurich

More of SPCL's research:

 youtube.com/@spcl **210+ Talks**

 twitter.com/spcl_eth **1.6K+ Followers**

 github.com/spcl **5,5K+ Stars**

... or spcl.ethz.ch



spcl/PraaS



Paper