

The Next Service Wave: Prototyping Cloud-Native and Stealthy Applications

Josef Spillner <josef.spillner@zhaw.ch>
Service Prototyping Lab (blog.zhaw.ch/icclab)

Sep 9, 2015 | IBM Research Zürich

Who we are

Service Prototyping Lab @ ZHAW School of Engineering / InIT, Winterthur
- SPLab in conjunction with ICCLab



Pietro Brossi



Özgür Özsü



Dr. Josef Spillner



Christof Marti



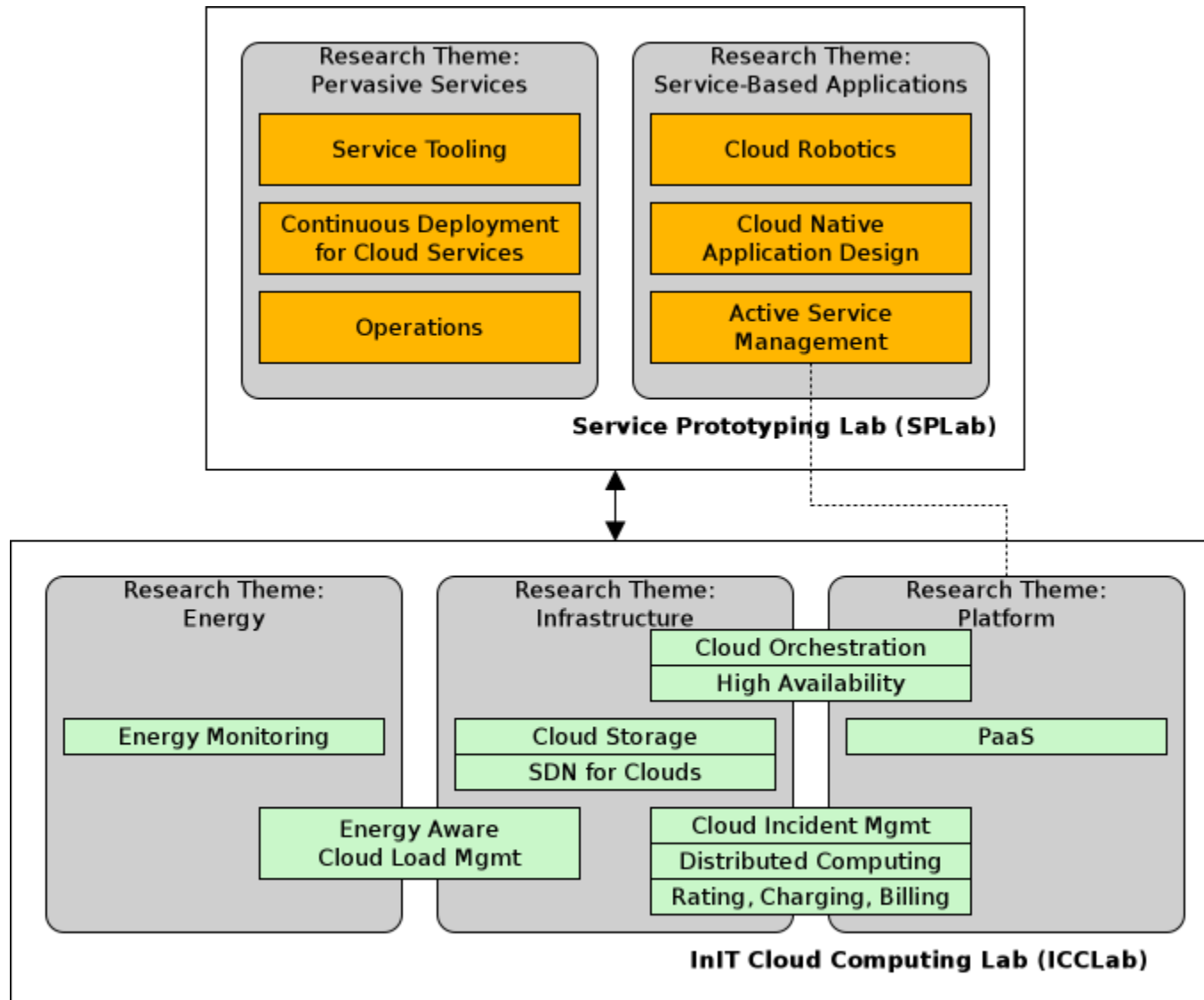
Dr. Giovanni Toffetti Carughi



We are hiring :)

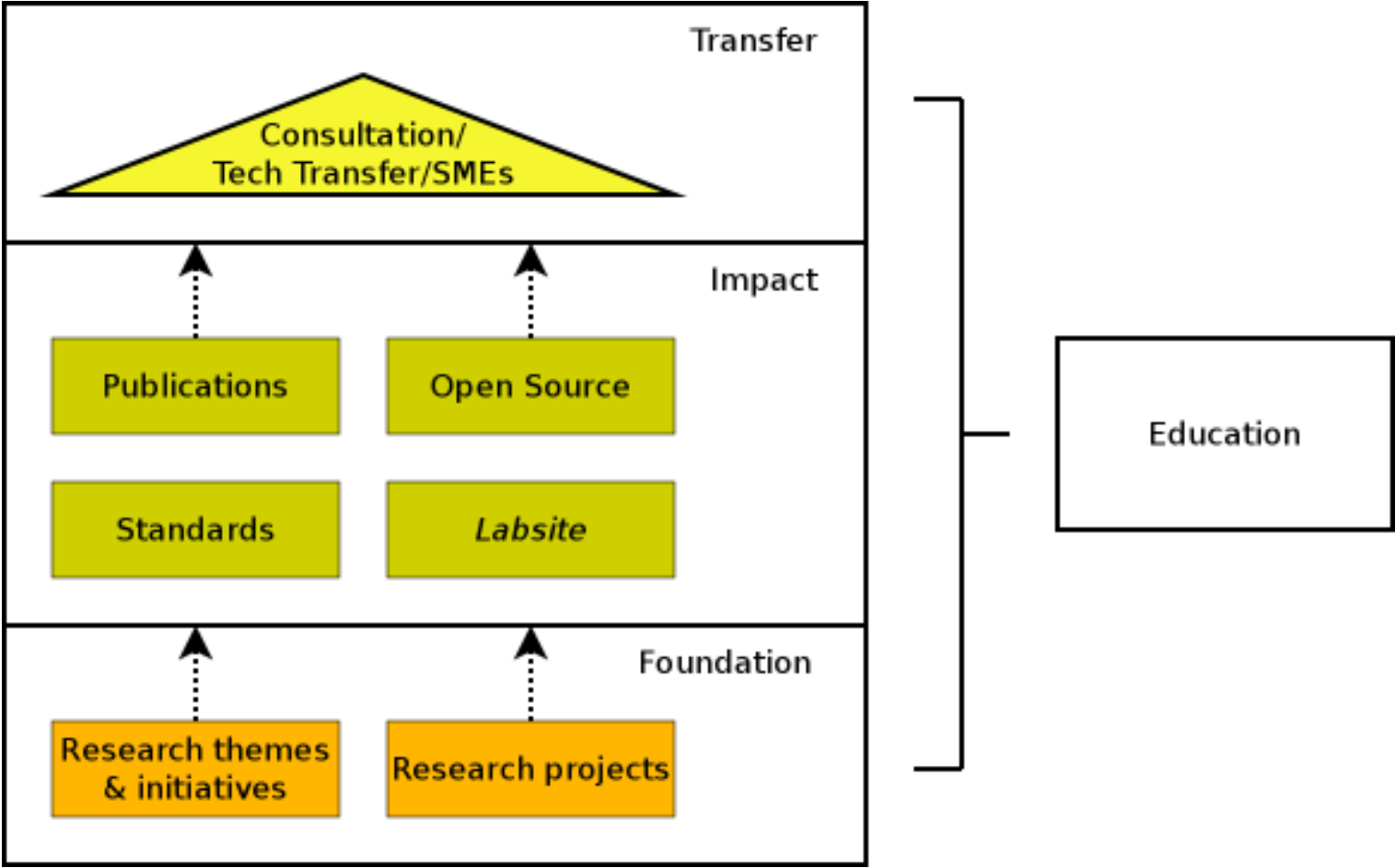
What we do

Themes, initiatives and projects



How we work

Fairly lab-style...



More information on SPLab...

Website: <http://blog.zhaw.ch/icclab/>

Microblog: Twitter @S_P_Lab

A Blog of the ZHAW Zurich University of Applied Sciences



ICCLAB SPLAB

About Us Imprint

Search

SUBSCRIBE

JOIN US
[Open positions](#)

SERVICE ENGINEERING

- Articles
 - Charging
 - CloudFoundry
 - Colloquium
 - Community
 - Education
 - Events
 - HowTo

RECENT POSTS

Announcing SCALE-UP Project

Posted on 7. September 2015 by Piyush Harsh

ICCLab, ZHAW is pleased to announce the successful start of the project [SCALE-UP](#) which is funded under program CUS 2013-2016 P-2 from [swissuniversities](#). The project consortium includes 9 leading universities of Switzerland including EPFL, Universities of Bern, Basel, St. Gallen, UST, ZHAW, FHNW, FFHS, and FHSG. The kick-off meeting took place on August 28, 2015, at the project coordinator's premises. The consortium is led by SWITCH. [Continue reading →](#)

Posted in [Articles](#), [Charging](#) | [Leave a reply](#) (0)

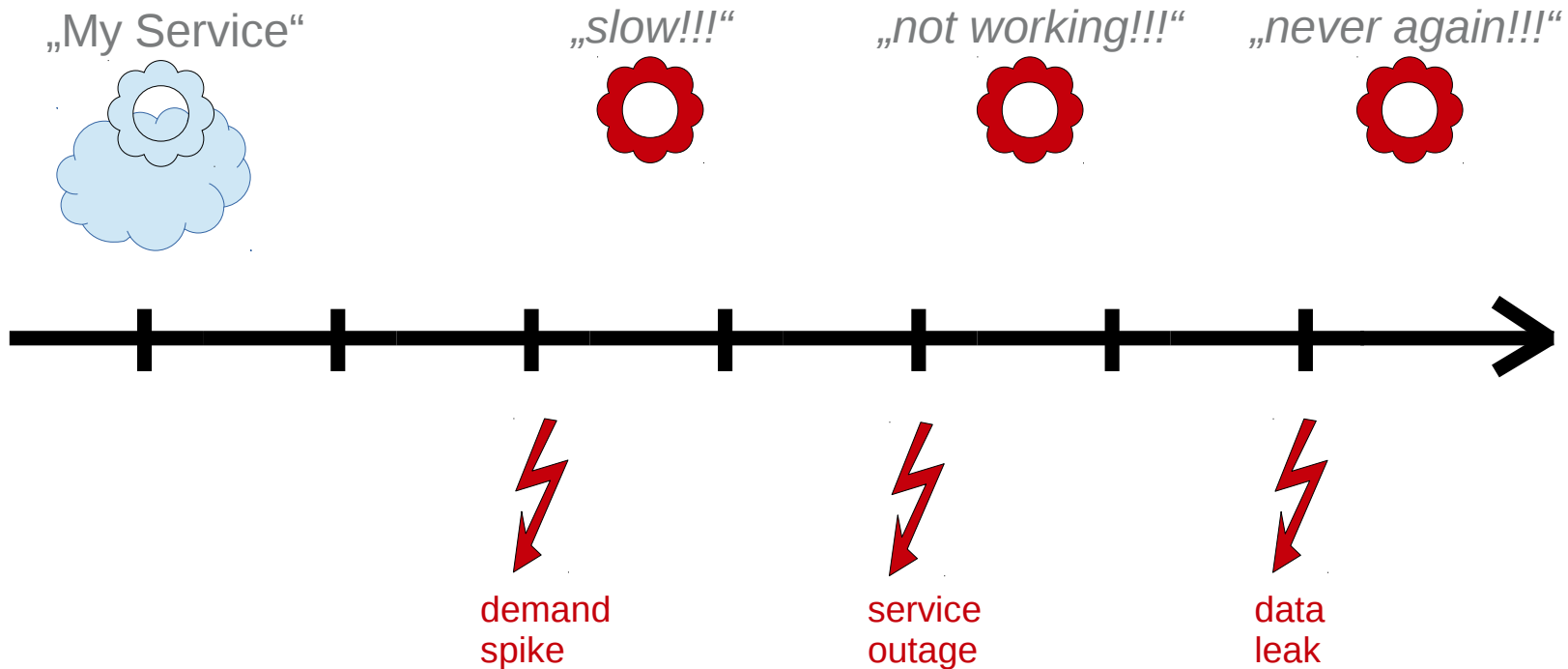
Announcing the Service Prototyping Lab [LEAVE A REPLY](#)

Service Prototyping Lab (SPLab) [LEAVE A REPLY](#)



Cloud-native & stealthy applications

Motivation: Today's cloud applications and services



Analysis of protection goals and risks

Goals:

- availability
- confidentiality
- scalability
- reliability
- resilience

Risks:

- unavailability of dependency service
- permanent exit (e.g. business termination, bankruptcy)
- condition changes (e.g. price jumps)
- loss of data (& backup, too)
- leak of data

Constraints:

- re-usable solution

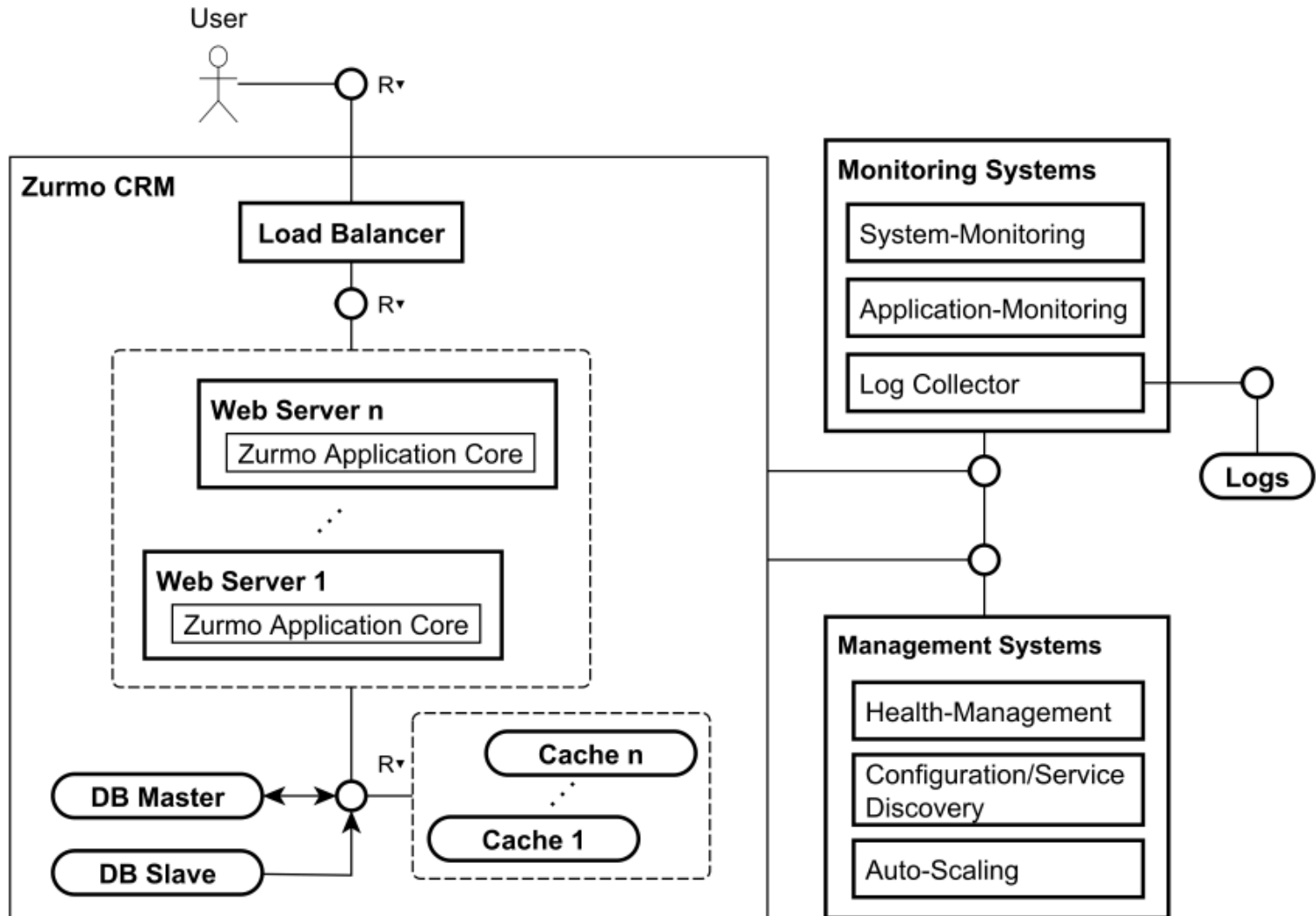
Cloud-native applications

Protection goals: Resilience and scalability

Focus: applications subdivided into microservices

Approach: Distributed consensus, container deployer and autoscaler

Cloud-native applications



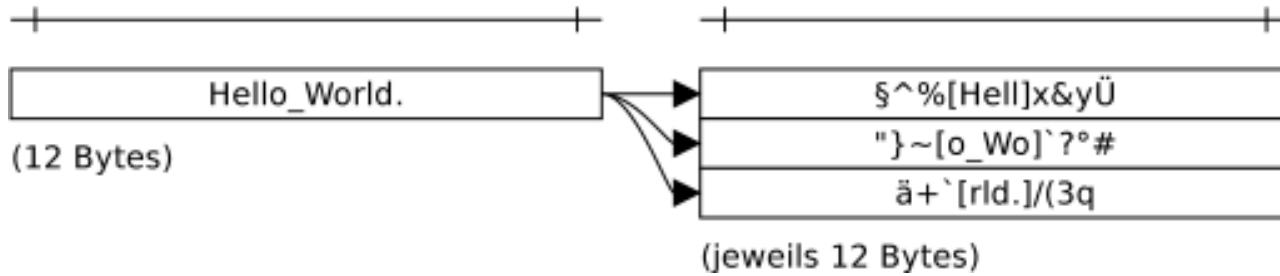
Stealth applications

Protection goals: Availability and confidentiality

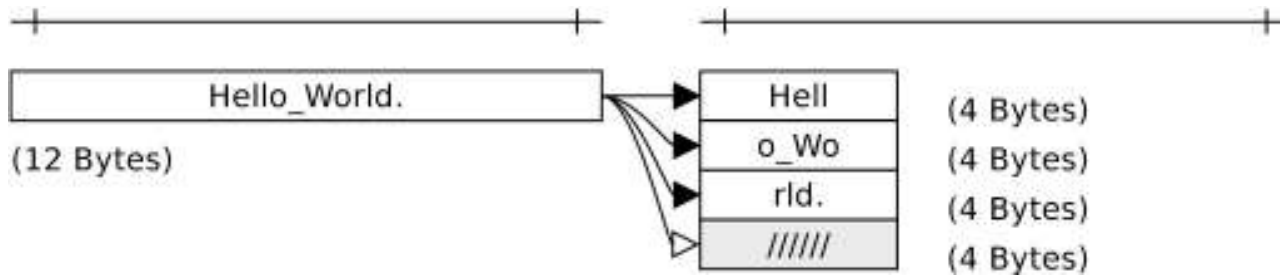
Focus: data-processing applications

Approach: Operation-aware data coding, distribution and processing
(in one multi-service pipeline)

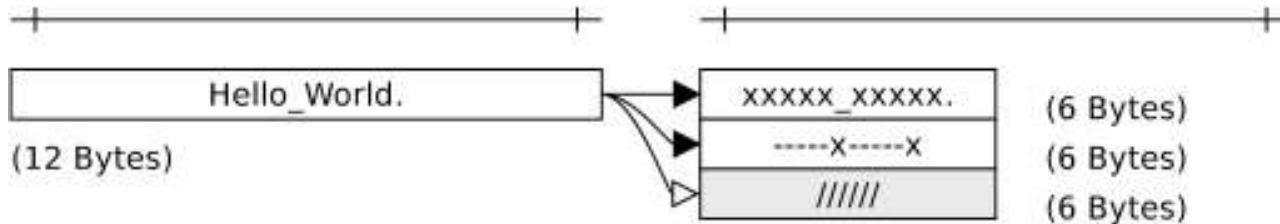
Data coding



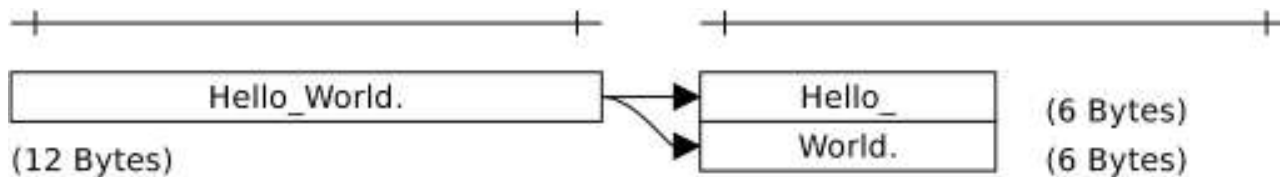
secret sharing



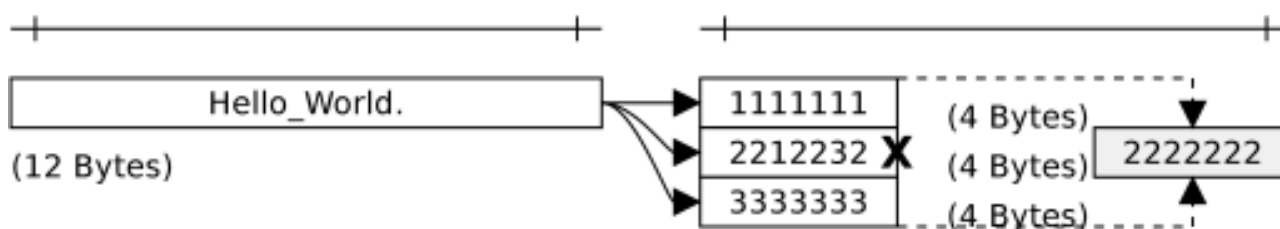
erasure coding
replication



bit splitting

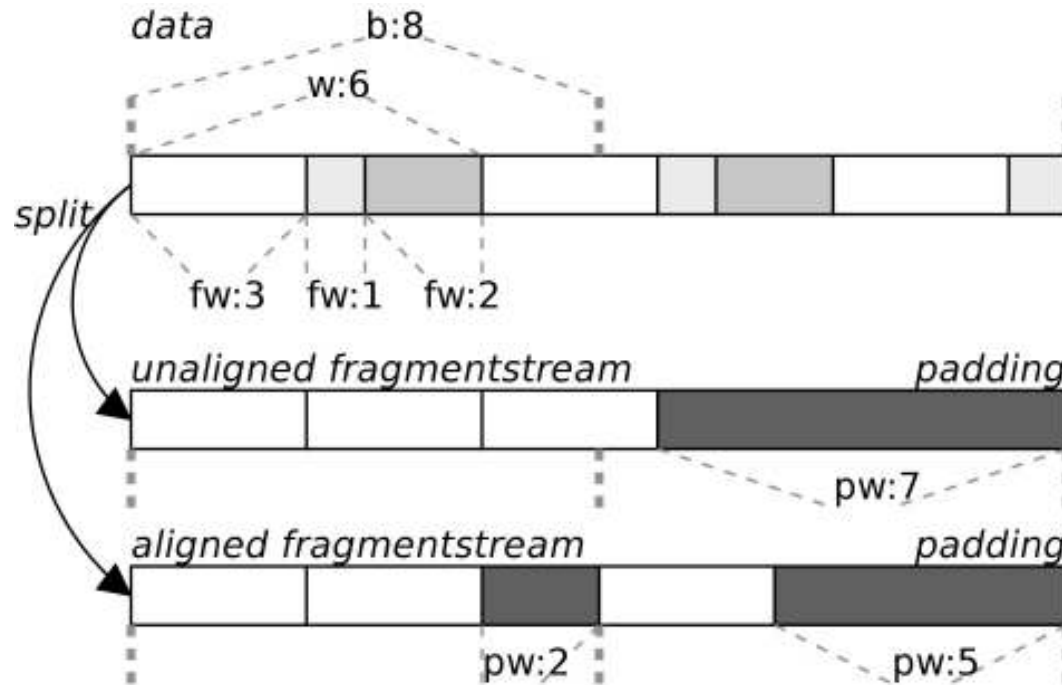


chunking



interpolation

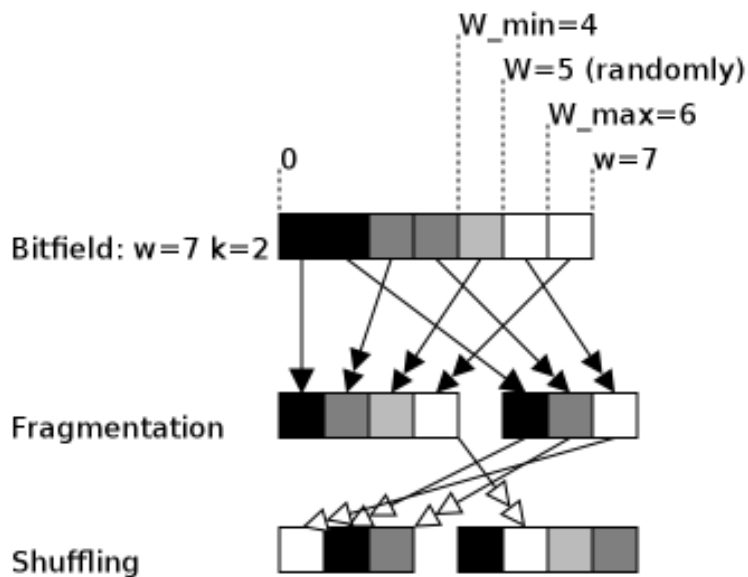
Data coding: Bitsplitting



Implications for the application author:

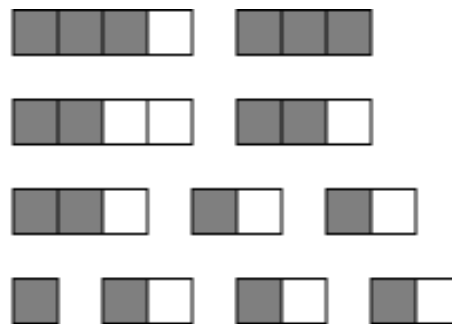
- should not need to worry
- advanced tooling needed

Data coding: Secret Sharing

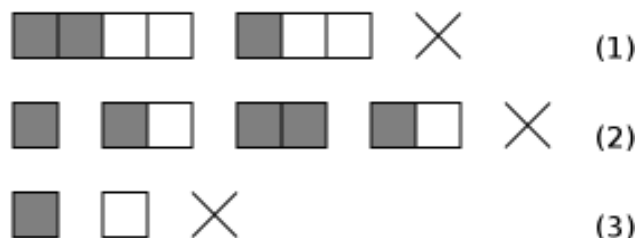


- Fixed assignment
- Rotated assignment
- ▷ Random assignment

Valid variants



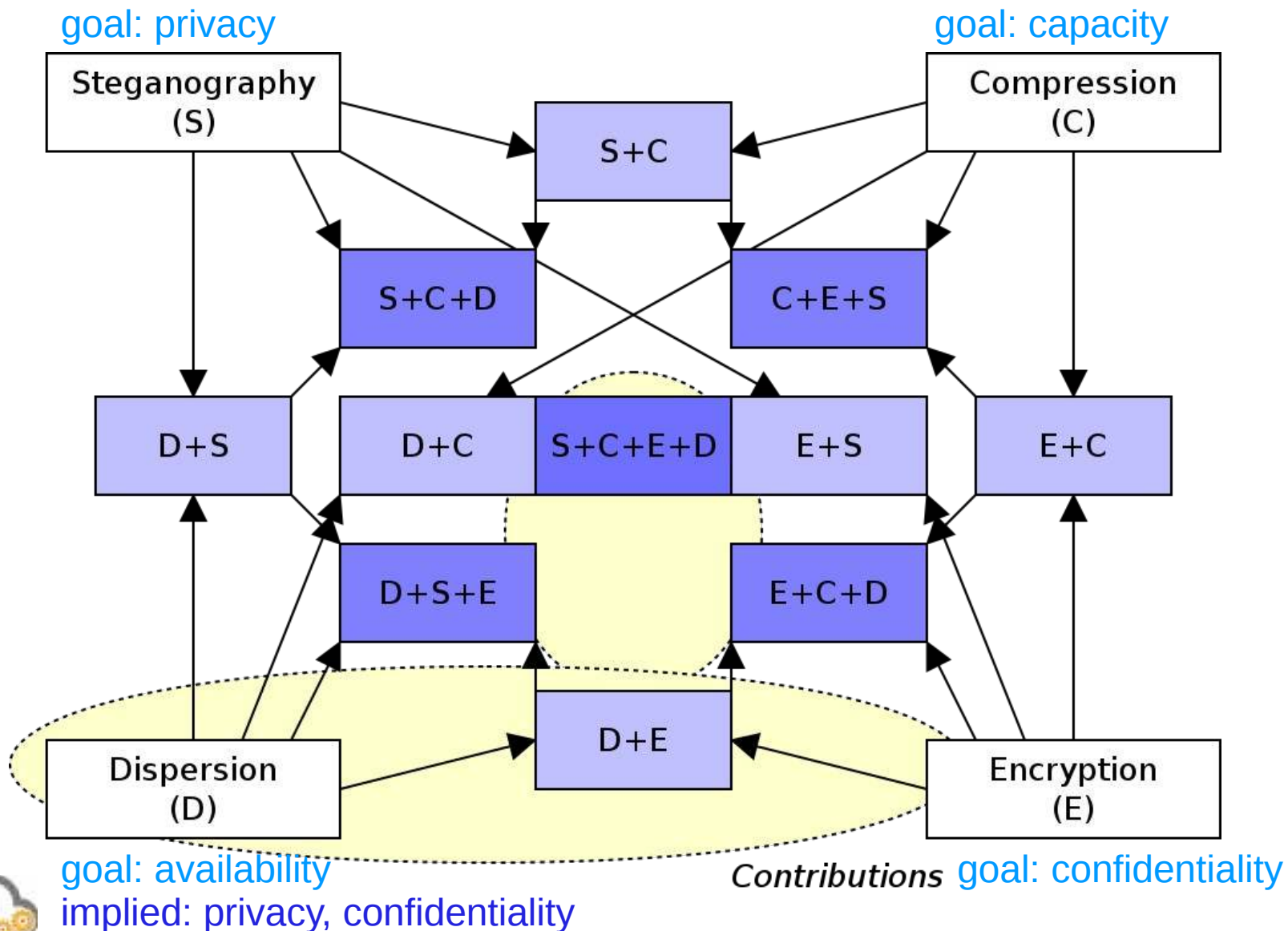
Invalid variants



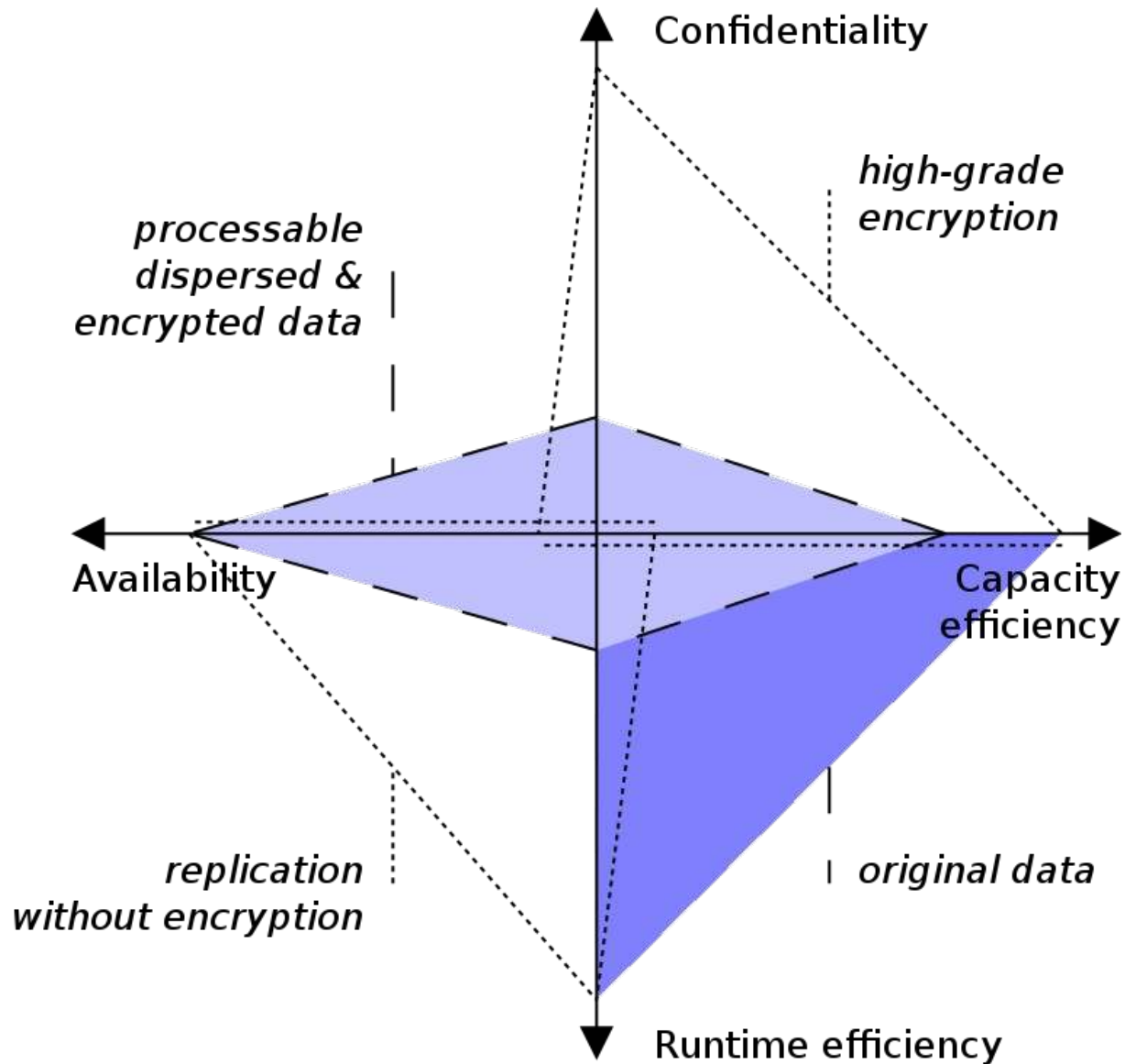
Error cases:

- (1) one bit too few set, Hamming weight wrongly determined
- (2) one bit too much set, secret-sharing rule not honoured
- (3) non-decidable variant and fragment without information

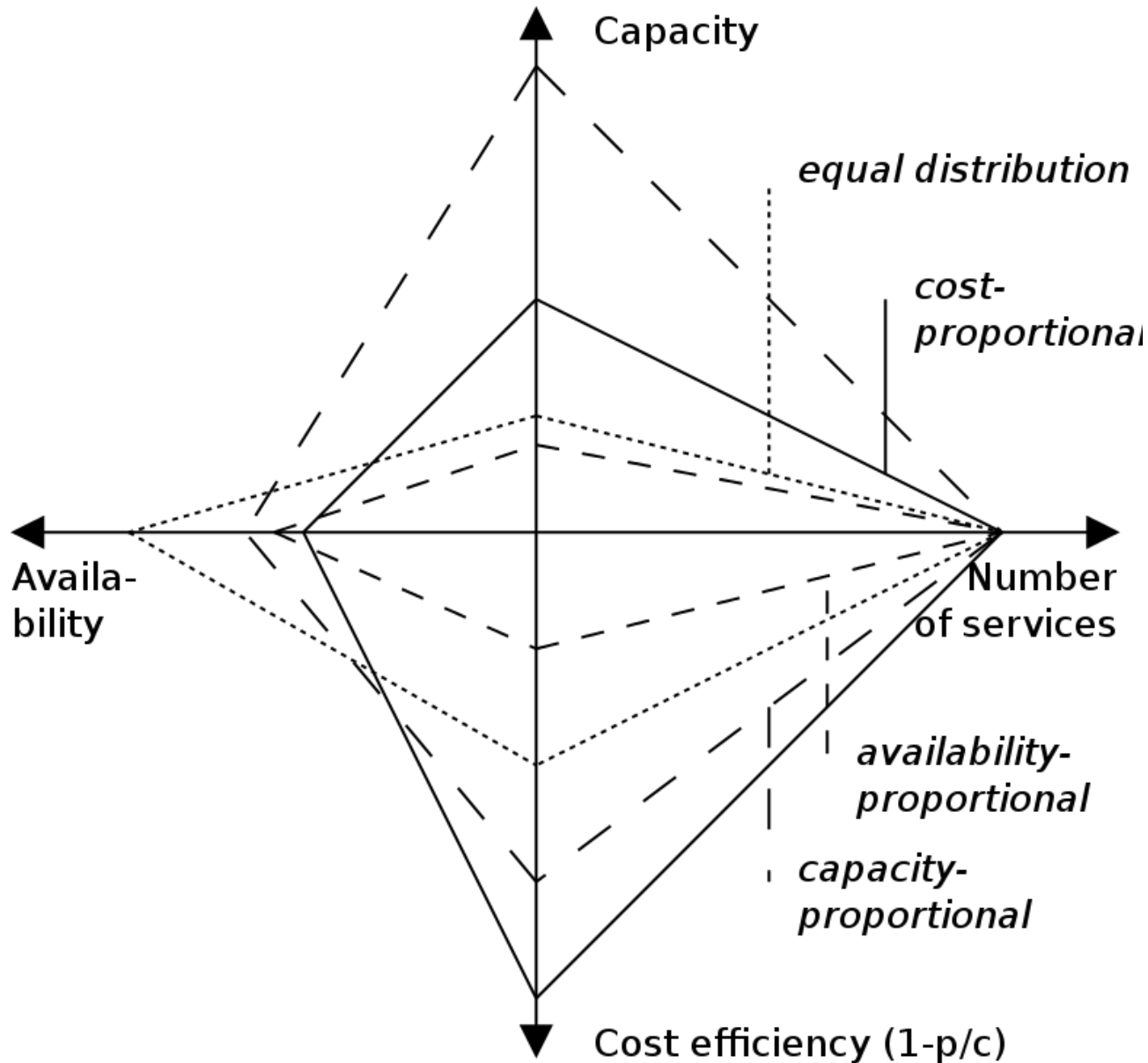
Combined („stealthy“) data coding



Data coding trade-offs

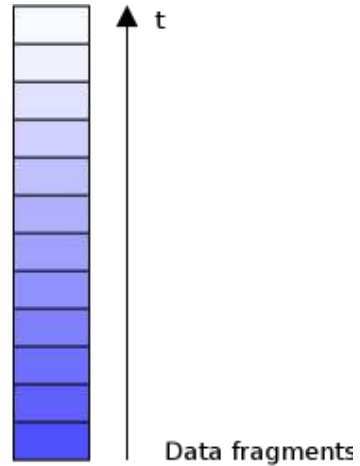
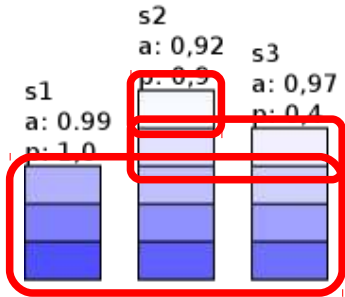


Data distribution trade-offs

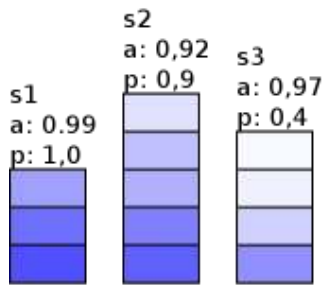


Data distribution strategies/algorithms

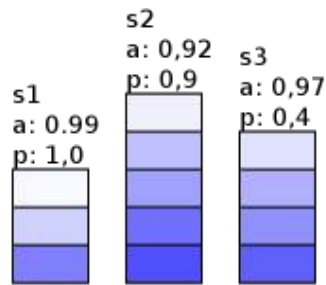
Equal distribution (for secret sharing)



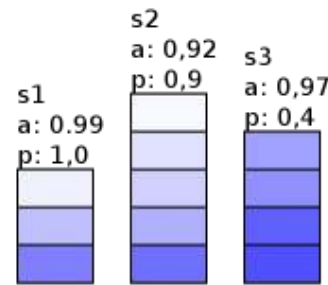
Proportional distribution [a]



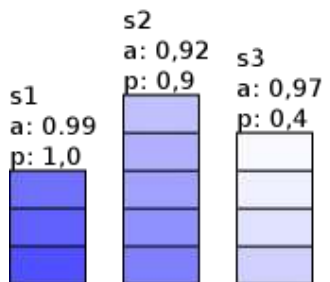
Proportional distribution [c]



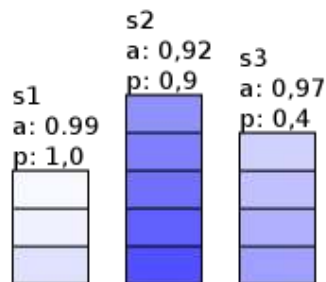
Proportional distribution [p]



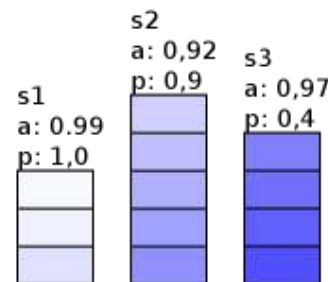
Absolute distribution [a]



Absolute distribution [c]



Absolute distribution [p]



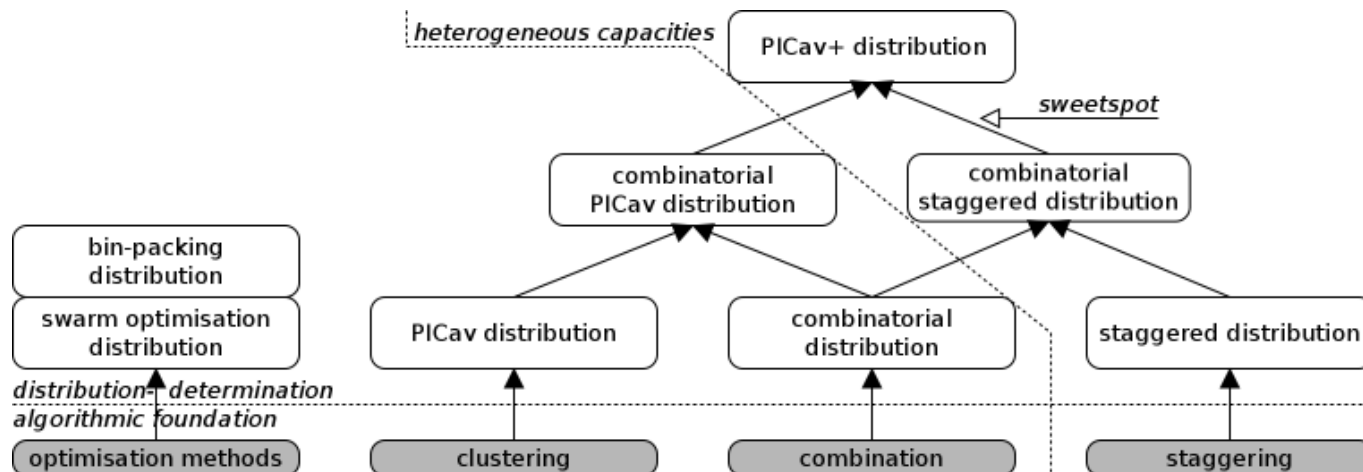
PICav+: The best-of-breed strategy

- powerful: optimises for capacity, price, availability constraints & runtime
- staggered: considers all elements in powerset of candidate service set
- sliced: capacity-maximising calculation rings
- iterative: finds some result first, finds best result eventually
- fast (clustering) & precise

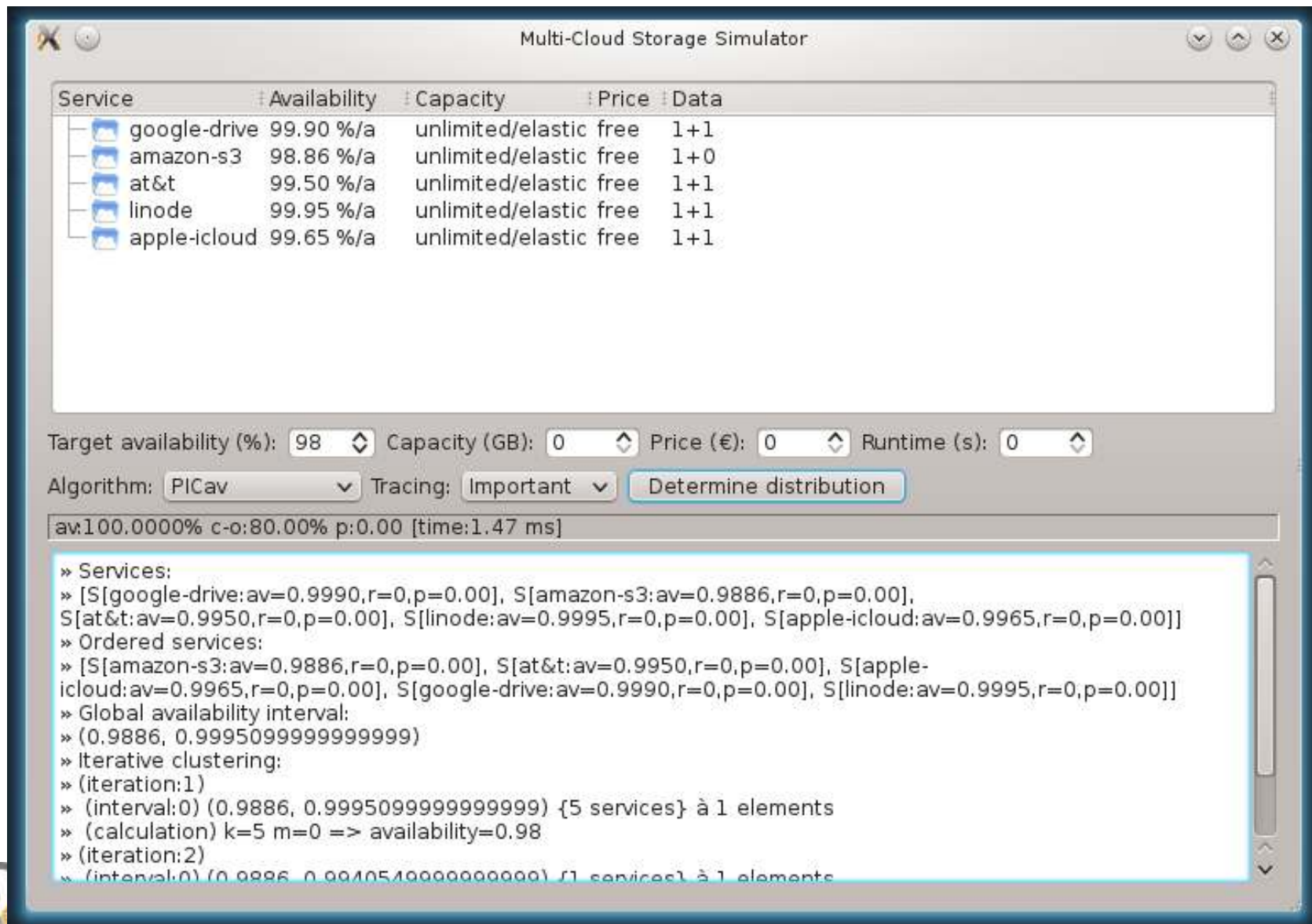
For each slice:

Homogeneous complexity: $availability = \sum_{i=1}^n \binom{n}{i} a_1^i (n - a_1)^{n-i}$

Heterogeneous complexity: $availability = \sum_{S \in P_{>k}(C)} \left(\prod_{i \in S} a_i \cdot \prod_{i \in C \setminus S} 1 - a_i \right)$



Data distribution tool: MCS-SIM/EMU



Multi-Cloud Storage Simulator

Service	Availability	Capacity	Price	Data
google-drive	99.90 %/a	unlimited/elastic	free	1+1
amazon-s3	98.86 %/a	unlimited/elastic	free	1+0
at&t	99.50 %/a	unlimited/elastic	free	1+1
linode	99.95 %/a	unlimited/elastic	free	1+1
apple-icloud	99.65 %/a	unlimited/elastic	free	1+1

Target availability (%): 98 Capacity (GB): 0 Price (€): 0 Runtime (s): 0

Algorithm: PICav Tracing: Important **Determine distribution**

av:100.0000% c-o:80.00% p:0.00 [time:1.47 ms]

```
» Services:
» [S[google-drive:av=0.9990,r=0,p=0.00], S[amazon-s3:av=0.9886,r=0,p=0.00],
S[at&t:av=0.9950,r=0,p=0.00], S[linode:av=0.9995,r=0,p=0.00], S[apple-icloud:av=0.9965,r=0,p=0.00]]
» Ordered services:
» [S[amazon-s3:av=0.9886,r=0,p=0.00], S[at&t:av=0.9950,r=0,p=0.00], S[apple-icloud:av=0.9965,r=0,p=0.00], S[google-drive:av=0.9990,r=0,p=0.00], S[linode:av=0.9995,r=0,p=0.00]]
» Global availability interval:
» (0.9886, 0.9995099999999999)
» Iterative clustering:
» (iteration:1)
» (interval:0) (0.9886, 0.9995099999999999) {5 services} à 1 elements
» (calculation) k=5 m=0 => availability=0.98
» (iteration:2)
» (interval:0) (0.9886, 0.9940549999999999) {1 services} à 1 elements
```

MCS-SIM results

```

Services: T1,T10,T2,T3,T4,T5,T6,T7,T8,T9
Distribution [algorithm: fixed[sp] time: 41.49]: {1+0,1+0,1+0,1+0,1+0,1+0,1+0,1+0,1+0,1+0} error, no solution found; discarding availability=0.565
Distribution [algorithm: fixed[re] time: 39.86]: {1+0,0+1,0+1,0+1,0+1,0+1,0+1,0+1,0+1,0+1} availability=1.0000 price=10.00 capacity-overhead=0.90
Distribution [algorithm: proportional[av] time: 39.66]: {0+0,0+0,1+0,1+0,1+0,1+0,0+0,0+0,0+0,0+0} error, no solution found; discarding availability=0.8583
Distribution [algorithm: proportional[ca] time: 39.80]: {1+0,1+0,1+0,1+0,1+0,1+0,1+0,1+0,1+0,1+0} error, no solution found; discarding availability=0.5653
Distribution [algorithm: proportional[pr] time: 39.75]: {0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0} error, no solution found; discarding availability=0.9100
Distribution [algorithm: absolute[av] time: 39.72]: {10+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0} availability=0.9900 price=1.00 capacity-overhead=0.00
Distribution [algorithm: absolute[ca] time: 39.94]: {10+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0} availability=0.9900 price=1.00 capacity-overhead=0.00
Distribution [algorithm: absolute[pr] time: 39.79]: {10+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0} availability=0.9900 price=1.00 capacity-overhead=0.00
Distribution [algorithm: random time: 40.05]: {1+0,1+1,0+0,1+1,1+0,1+0,0+0,1+1,1+0,0+0} availability=0.9582 price=7.00 capacity-overhead=0.30
Distribution [algorithm: combinatory time: 300.17]: {1+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0,0+0} availability=0.9900 price=1.00 capacity-overhead=0.00
Distribution [algorithm: staggered[pl] time: 386.44]: {1+x,1+x,1+x,1+x,1+x,1+x,1+x,1+x,1+x,1+x} error, solution found but runtime exceeded by 0.09s
Distribution [algorithm: staggered[co] time: 301.59]: {1+x,1+x,1+x,1+x,0+x,0+x,0+x,0+x,0+x,0+x} availability=0.9931 price=4.00 capacity-overhead=0.75
Distribution [algorithm: picav time: 78.41]: {1+1,1+0,1+1,1+1,1+1,1+1,1+0,1+0,1+0,1+0} availability=0.9993 price=10.00 capacity-overhead=0.50
Distribution [algorithm: picav+[av] time: 300.35]: {1+1,1+0,1+1,1+1,1+1,0+1,0+0,0+0,0+0,0+0} availability=0.9996 price=5.00 capacity-overhead=0.50
Distribution [algorithm: picav+[ca] time: 301.73]: {1+1,1+0,1+1,1+1,1+1,0+1,0+0,0+0,0+0,0+0} availability=0.9996 price=5.00 capacity-overhead=0.50
Distribution [algorithm: picav+[pr] time: 300.38]: {1+1,1+0,1+1,1+1,1+1,0+1,0+0,0+0,0+0,0+0} availability=0.9996 price=5.00 capacity-overhead=0.50
    
```



```

fixed[sp] time: 41.49: {1+0,1+0,...,1+0} error, no solution available
fixed[re] time: 39.86: {1+0,0+1,...,0+1} availability=1.0000 price=10.00 capov=.
staggered[pl] time:386.44: {1+x,1+x,...,1+x} error, solution found but rt exceeded

proportional[av/ca/pr]
absolute[av/ca/pr]
random
combinatory
staggered[co] *
picav
picav+[av/ca/pr] *
} C_n ⊂ C_h ∈ P(C) = {{T_1}, {T_1,T_2}, ..., {T_1, ..., T_n}}
= 16 assignment algorithm variants, 4 combinatory x 3 staggered = 2 optim.
    
```



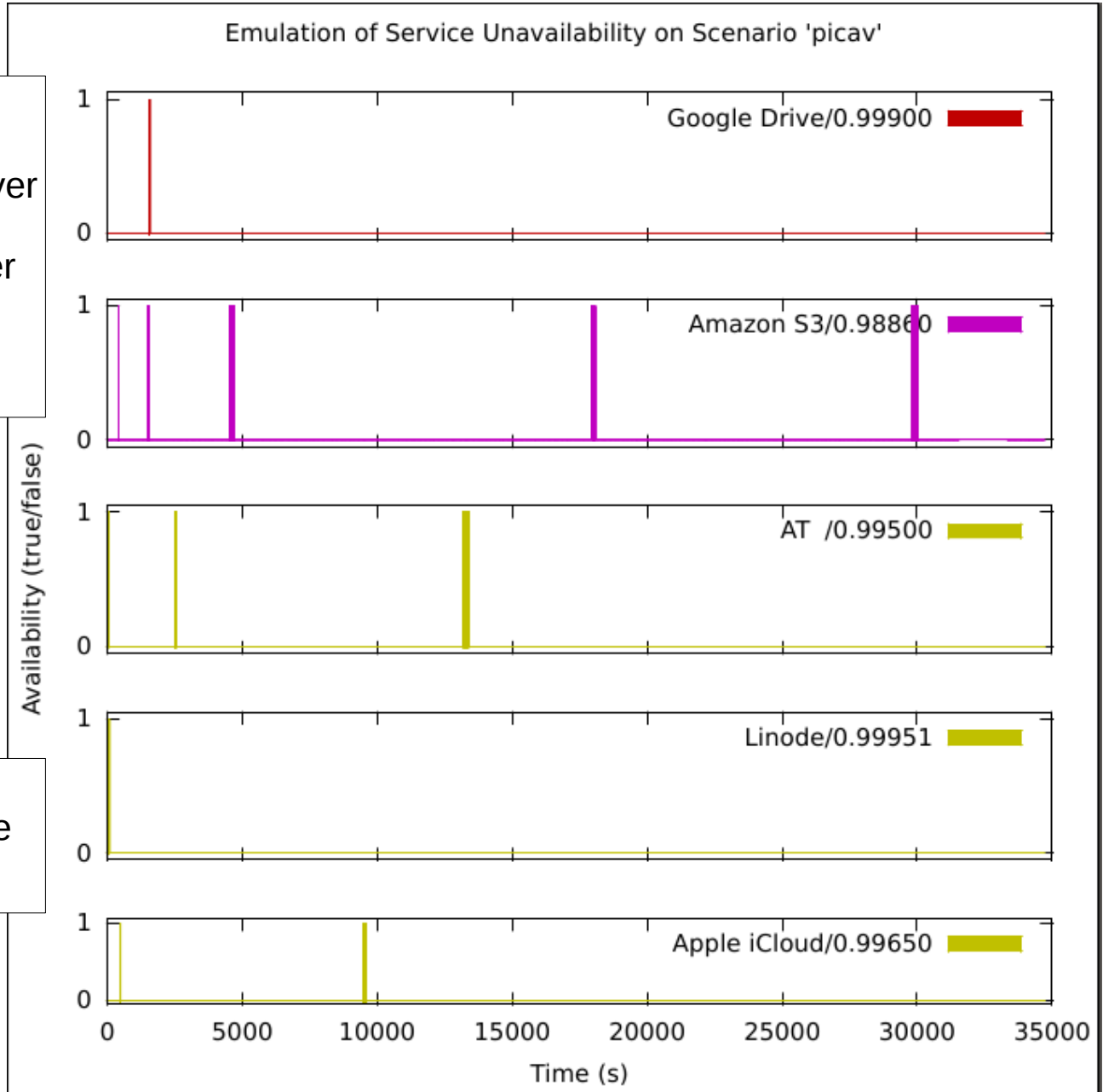
MCS-EMU results

4 targets:

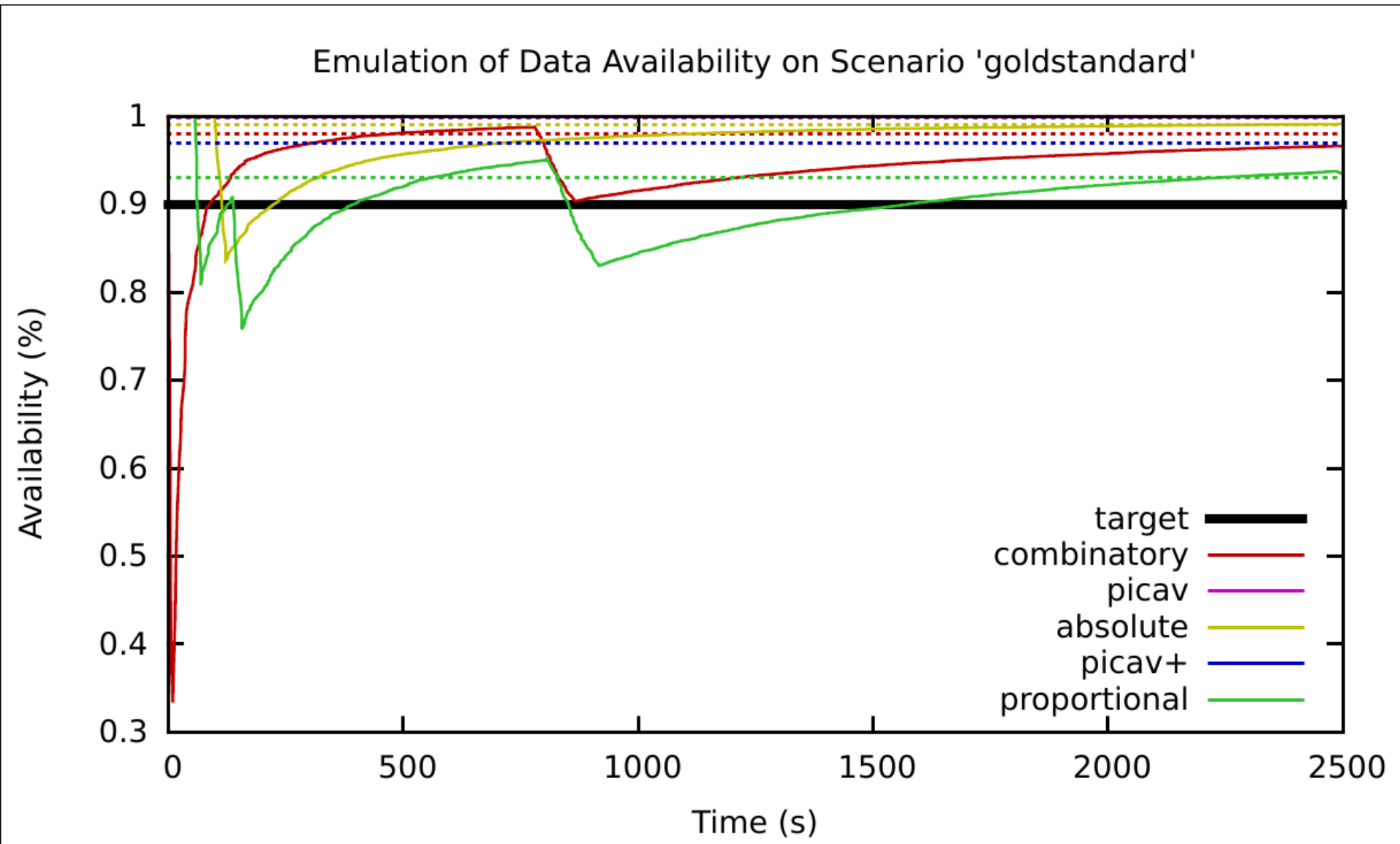
- no-op (sim.)
- web/file server (storage)
- OS container (compute)
- L4 proxy (network)

2 models:

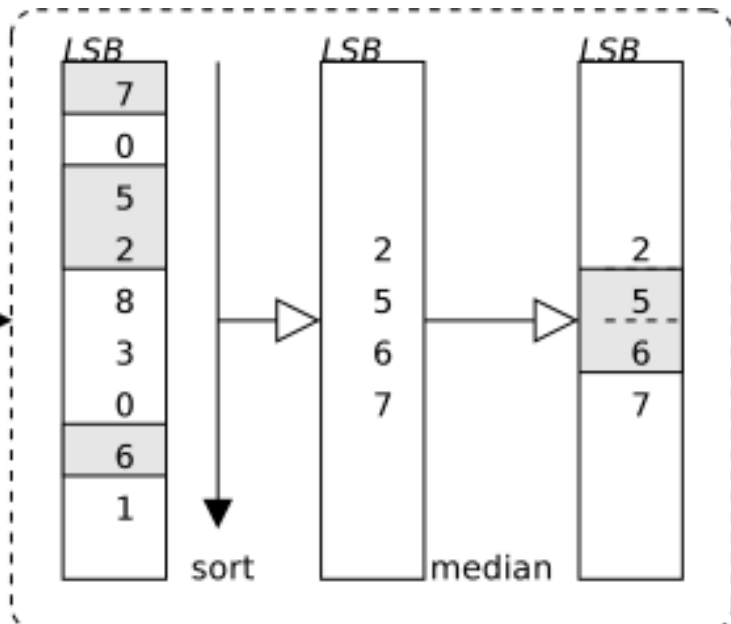
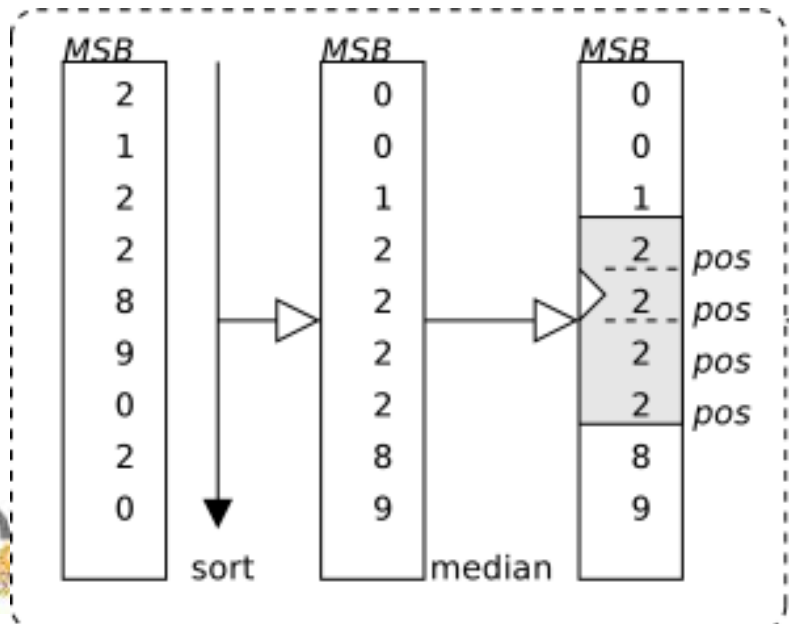
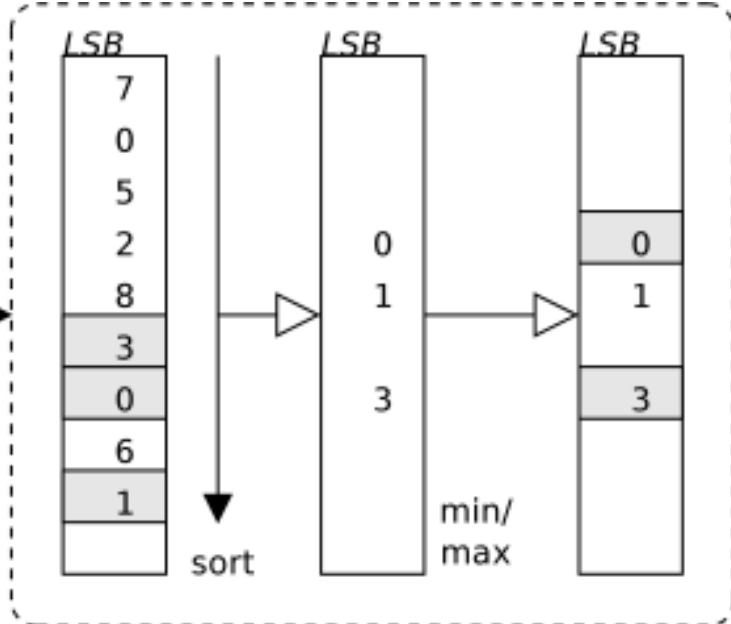
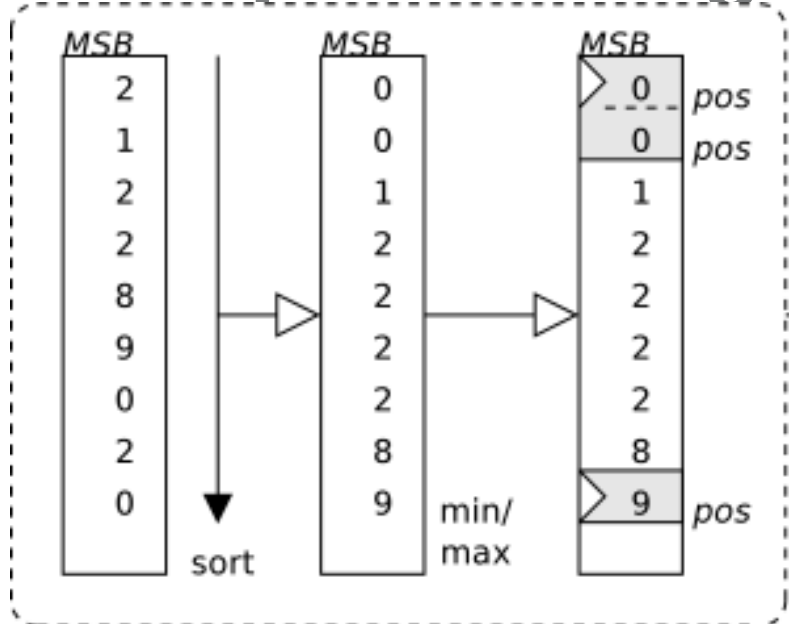
- convergence
- incident



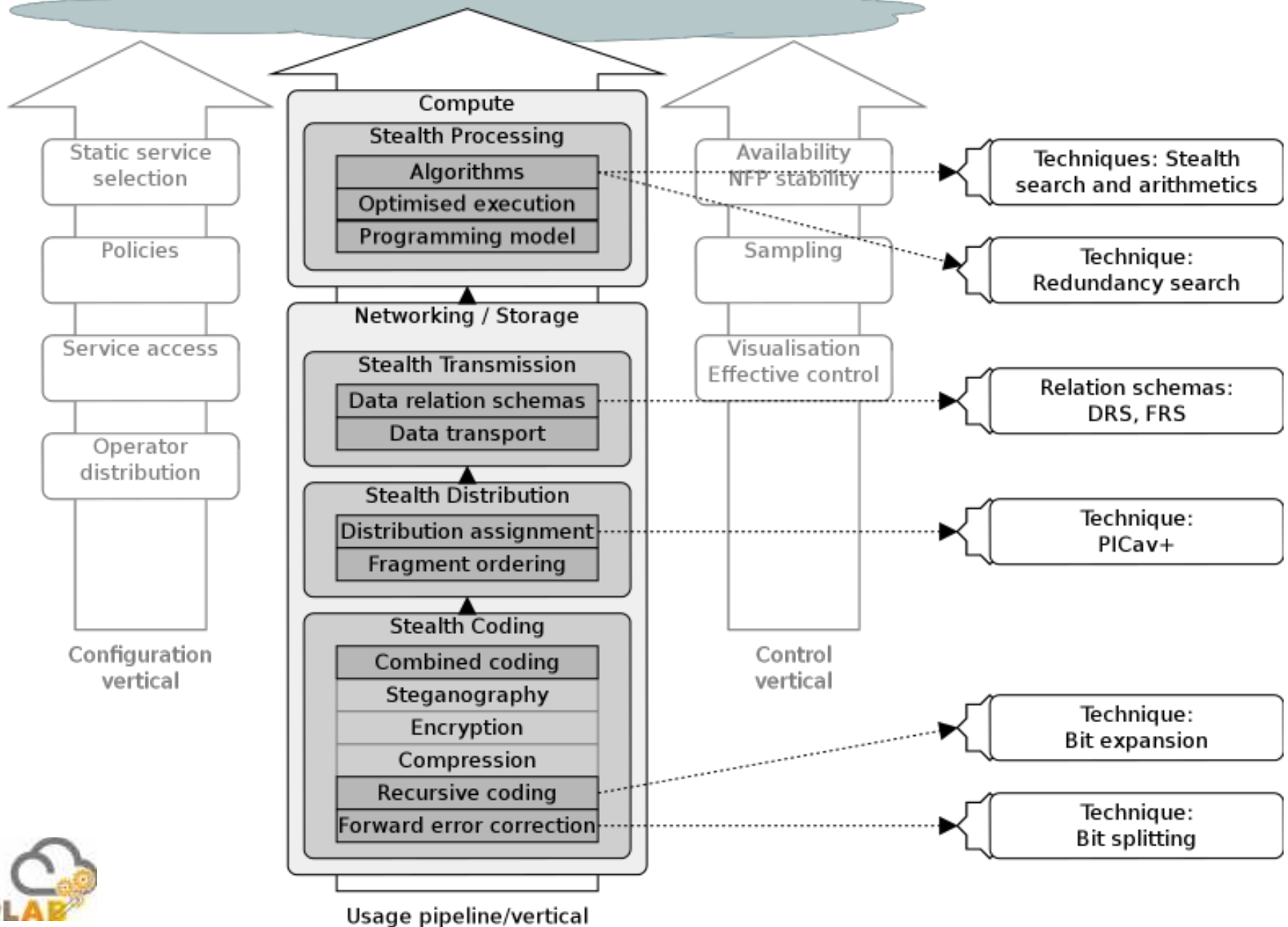
MCS-EMU results



Stealth processing

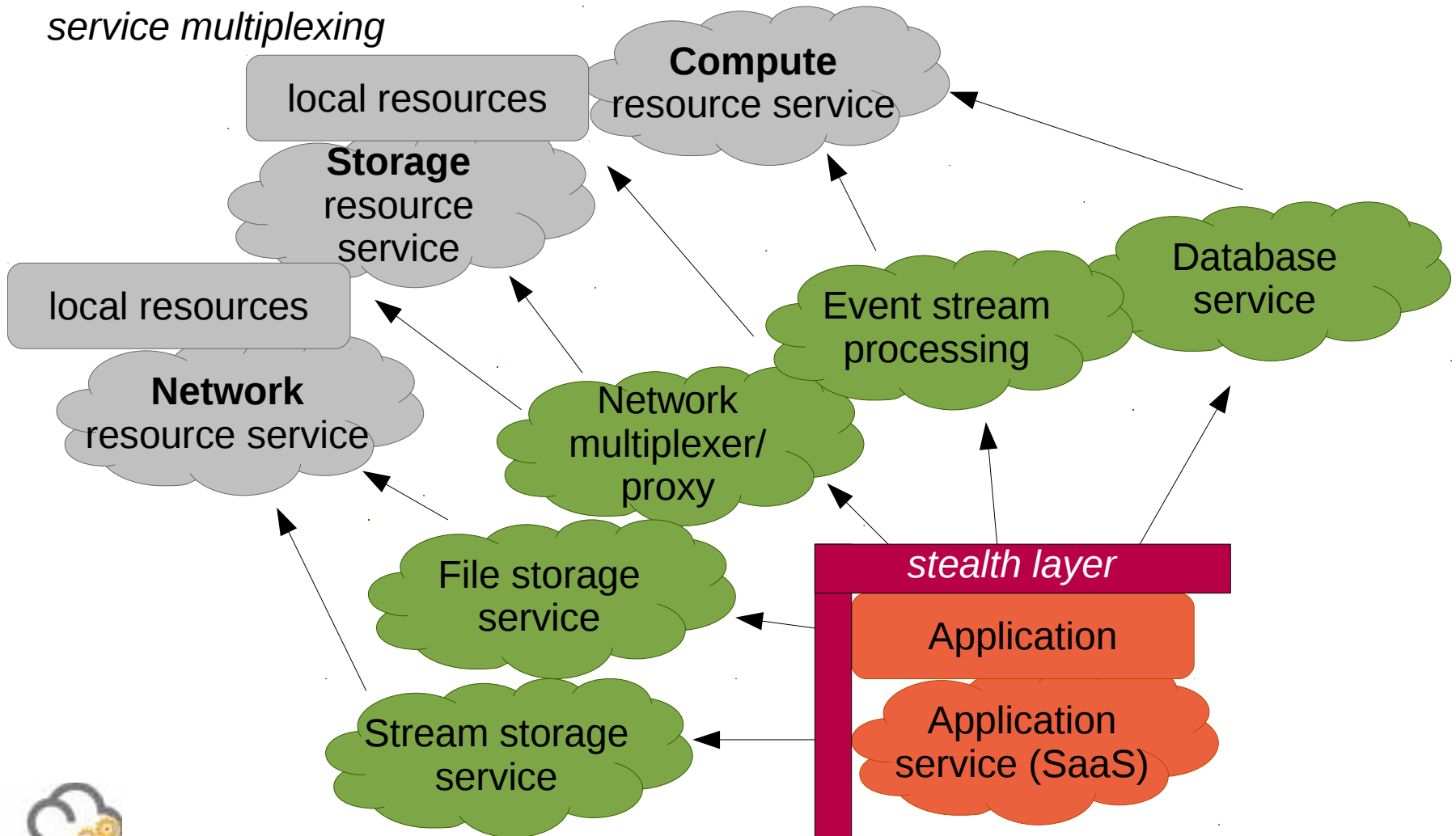


Stealth pipeline



Stealth layer

RMS: resource & service multiplexing



Stealth layer

[Image sources: dreamstime.com, bitrebels.com, suitssofar.com]

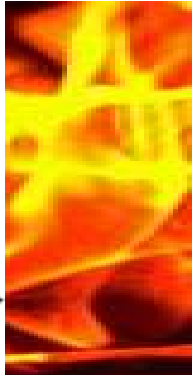
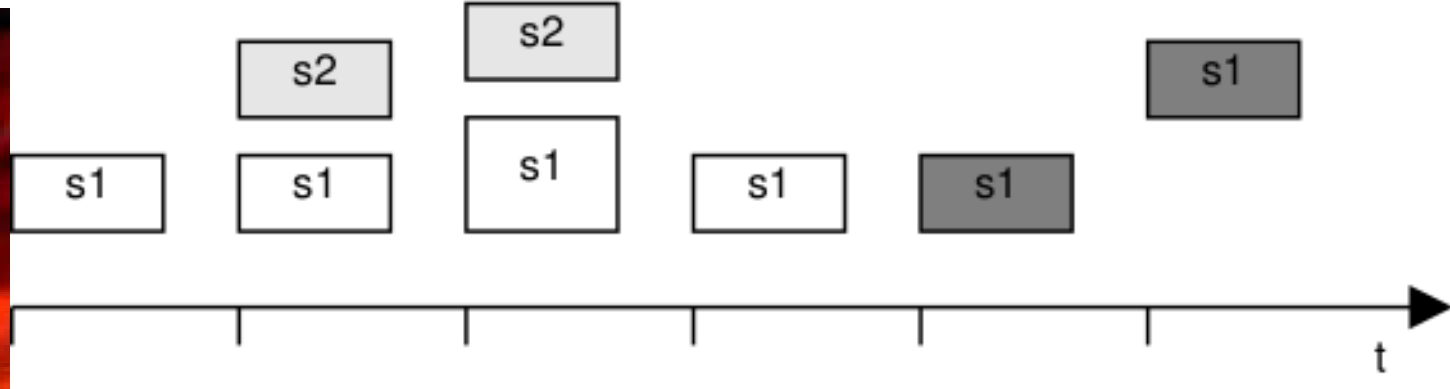


Stealth layer: Coverable cloud service evolution



initial addition growth decline & vanish price change location change

s: cloud services



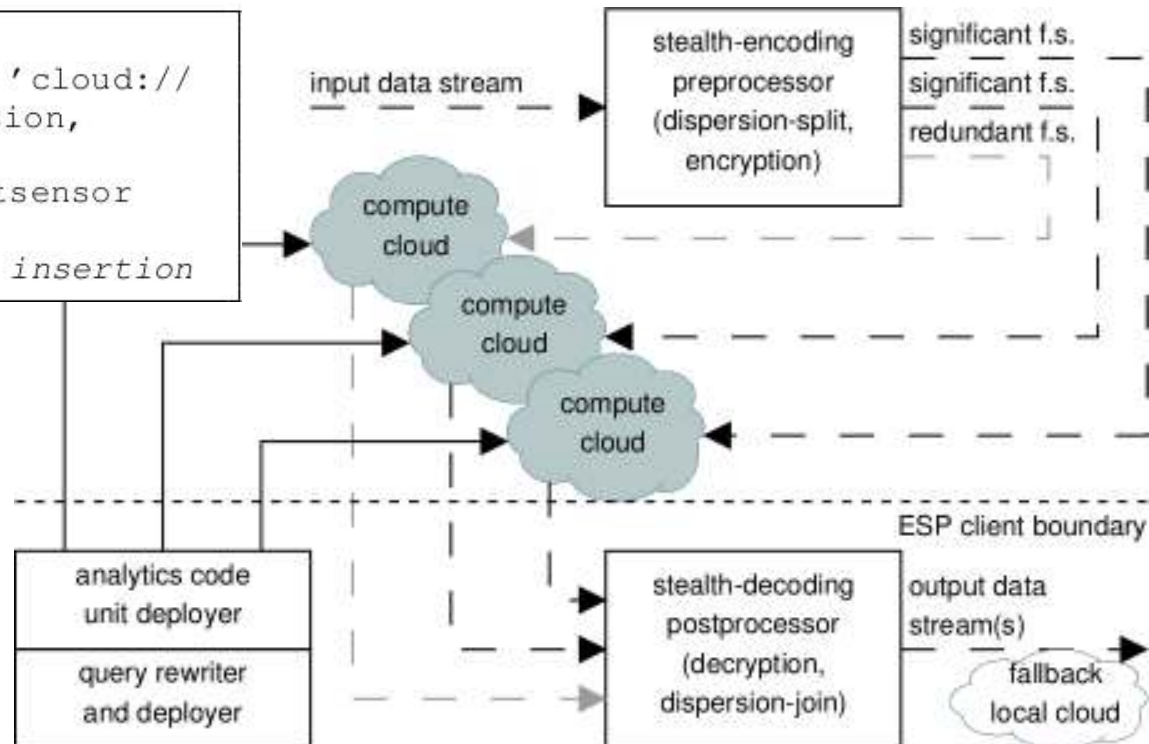
Showcase: StealthDB

Listing 1. Sender commands

```
\$ ./stealthdb sender
>>> USE CLOUDS 'cloud://cloud1' AND 'cloud://
cloud2' WITH 'encryption,dispersion,
ordered';
>>> CREATE TABLE heartsensor (frequency REAL);
>>> INSERT INTO heartsensor (frequency) VALUES
(102.4);
... -- more insertions follow
```

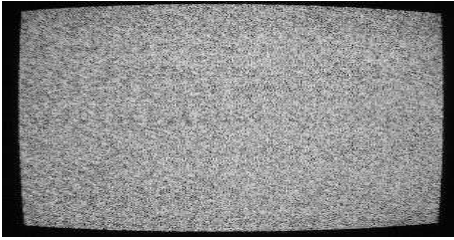
Listing 2. Receiver commands

```
\$ ./stealthdb receiver
>>> USE CLOUDS 'cloud://cloud1' AND 'cloud://
cloud2' WITH 'encryption,dispersion,
ordered';
>>> SELECT AVG(frequency) FROM heartsensor
WHERE frequency > 0.0 FOREVER;
... -- query gets triggered on each insertion
```



Showcase: StealthDB

A data fragment in the cloud...



... what can we do with it?

Features:

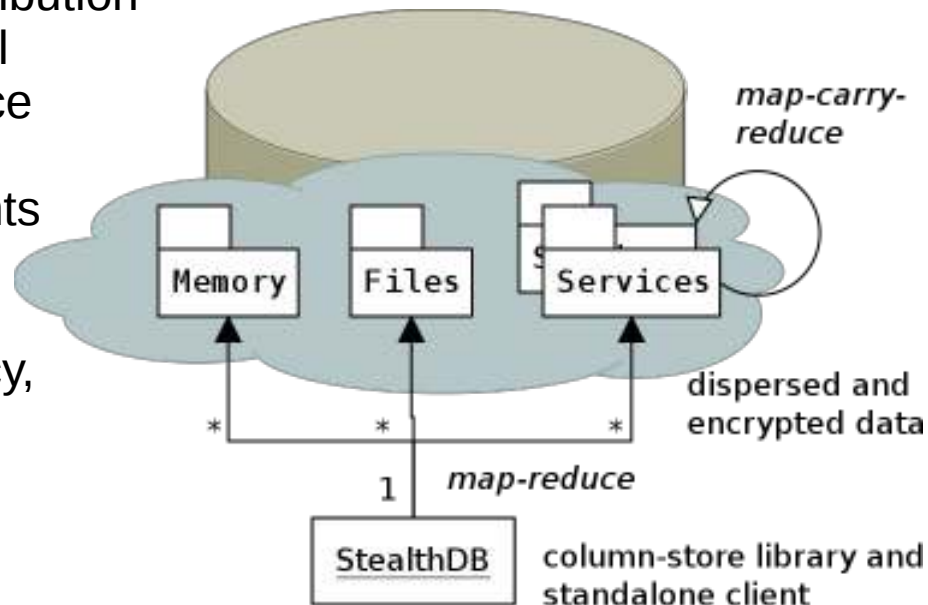
- per-column distribution
- migration control
- map-carry-reduce operations
- user requirements optimisation (performance, energy-efficiency, reliability, ...)

Dispersed Processing:

- structure-preserving bitsplitting
 - => search (any data)
 - => arithmetics, statistics (structured data)

Encrypted Dispersed Processing (Stealth Processing):

- homomorphic encryption
- order-preserving encryption
- searchable encryption



StealthDB in Action: overview

```
josef@rumba:/repos/space-universe/dispersedalgorithms/db$ ./stealthdb
~~ StealthDB >master >Wed May 20 16:14:37 2015 +0200 ~~
Type HELP; to get started.
Using database 'stealthdb'.
Storing all data and performing all procedures on ['mem://localhost'] with ['replication'].
>>> HELP;
StealthDB Quickhelp
HELP [<topic>]
SHOW DATABASES|TABLES
CREATE TABLE <table> [(<column> <column-type>, ...)]
DESCRIBE <table>
DROP TABLE [IF EXISTS] <table>
CREATE DATABASE <database>
USE DATABASE <database>
DROP DATABASE <database>
[EXPLAIN ANALYZE] SELECT [DISTINCT] */<column>/<aggregate>(*/<column>)/<predicate>, ... [FROM <table>]
  [WHERE <column> LIKE/=/.<value>] [ORDER BY <column> [ASC|DESC]] [OPTIMIZE FOR <goal>] [FOREVER]
INSERT INTO <table> (<column>, ...) VALUES (<value>, ...)
DELETE FROM <table>
USE CLOUDS <cloud> [AND <cloud>...][WITH <distribution>]
ALTER TABLE <table> [ALTER COLUMN <column>] USE CLOUDS ...
MODE <mode>
>>> █
```

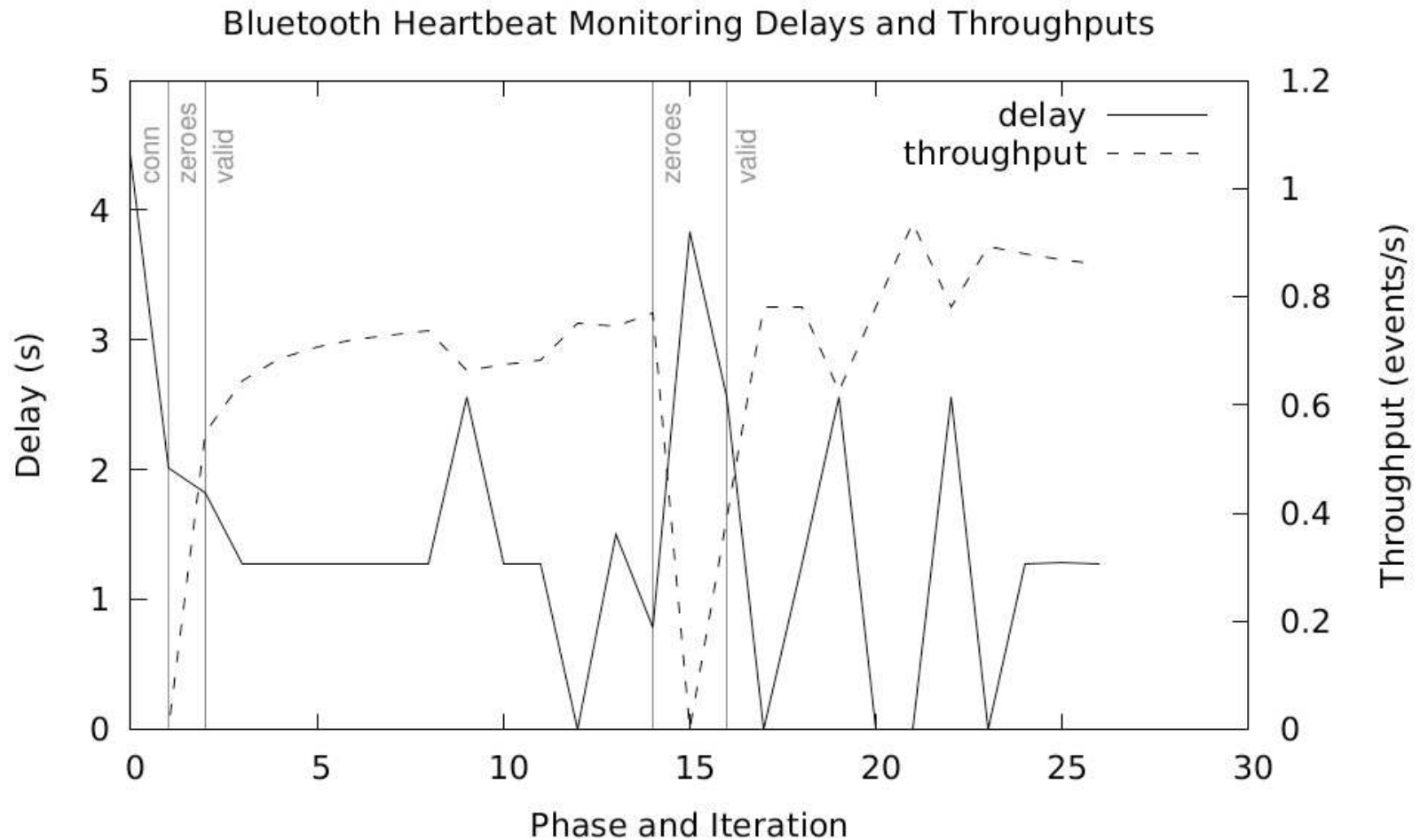
StealthDB in Action: multi-service

```
>>> USE CLOUDS 'mem://fastram' AND 'file:///globalfs/record' AND
... 'cloud://googleappengine-23' WITH 'hashring,encryption,ordered';
Using database 'stealthdb'.
Storing all data and performing all procedures on ['mem://fastram', 'file:///globalfs/record', 'cloud://googleappengine-23'] with ['hashring', 'encryption', 'ordered'].
>>> CREATE TABLE cloudsalary (person TEXT, chf INT);
Created table cloudsalary.
(DEBUG:notifier:watch /globalfs/record/cloudsalary/person)
(DEBUG:info=::info::stealthdb-cloud::googleappengine-23::)
Added column person of type TEXT.
(DEBUG:notifier:watch /globalfs/record/cloudsalary/chf)
(DEBUG:info=::info::stealthdb-cloud::googleappengine-23::)
Added column chf of type INT.
>>> █
```

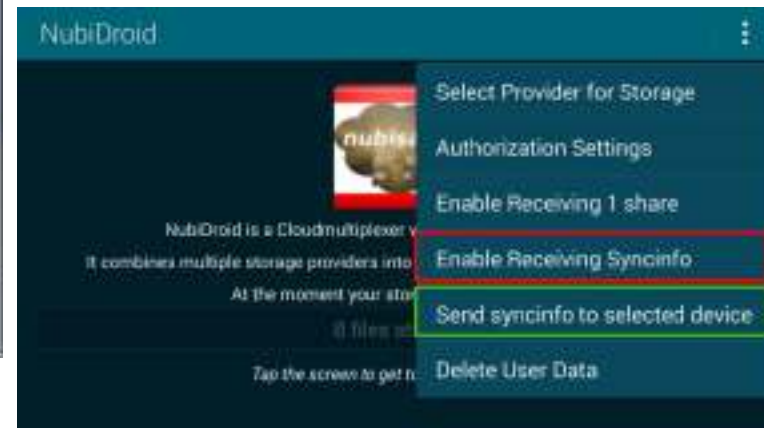
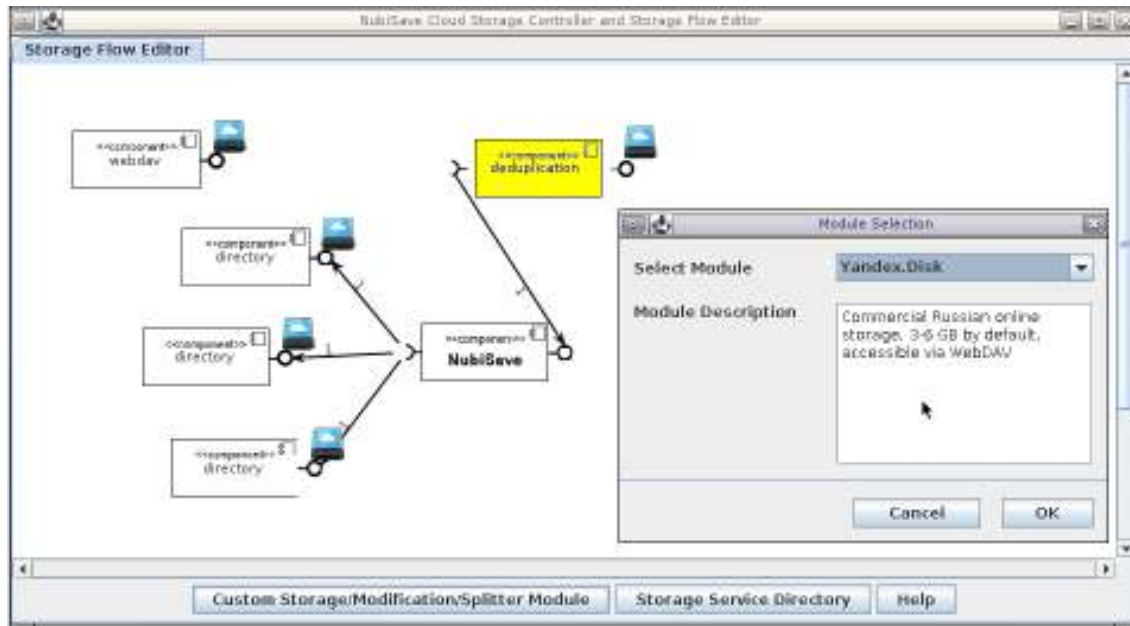

StealthDB in Action: protection goals

```
>>> SELECT MIN(chf) FROM cloudsalary OPTIMIZE FOR 'performance,precision';
(DEBUG:aggregate cloud:'mem://fastram')
(DEBUG:decryption:privkey=PrivateKey(l=276206633765143932185133659008305273600,m=77799637863792083604813961003219226636),pubkey=PublicKey(n=276206633765143932218448800529509760119,n_sq=76290104535872348142321164364999188369598776886438439468947400461280922894161,g=276206633765143932218448800529509760120),entry=74457922202228058251911302434069182833979044730019820823958984398650514208122)
(DEBUG:encryption:decrypt-safe)
(DEBUG:decryption:result=250000)
(DEBUG:decryption:privkey=PrivateKey(l=276206633765143932185133659008305273600,m=77799637863792083604813961003219226636),pubkey=PublicKey(n=276206633765143932218448800529509760119,n_sq=76290104535872348142321164364999188369598776886438439468947400461280922894161,g=276206633765143932218448800529509760120),entry=29962522632047879125545174731291639553194534076196749551474505932153721290035)
(DEBUG:encryption:decrypt-safe)
(DEBUG:decryption:result=180000)
(DEBUG:decryption:privkey=PrivateKey(l=276206633765143932185133659008305273600,m=77799637863792083604813961003219226636),pubkey=PublicKey(n=276206633765143932218448800529509760119,n_sq=76290104535872348142321164364999188369598776886438439468947400461280922894161,g=276206633765143932218448800529509760120),entry=40819492330637723794130390345172761293210144530235730019692667468160790251811)
(DEBUG:encryption:decrypt-safe)
(DEBUG:decryption:result=24999)
(DEBUG:aggregate sampling:1.0000%)
Minimum of chf: 24999
>>> █
```

Performance (depends on protection!)



File Service Multiplexing



DB Service Multiplexing

```
File "/opt/DBM+Hiwi-Repo/stealthdbtesting/StealthFlights/imp
return route.call(**args)
File "/opt/DBM+Hiwi-Repo/stealthdbtesting/StealthFlights/imp
rv = callback(*a, **ka)
File "todonew.py", line 108, in search city
testlist = flighttable.columns["city"].getentries()
File "/opt/DBM+Hiwi-Repo/stealthdbtesting/StealthFlights/imp
f = open(os.path.join(Cloud.url.path, self.tablename, self.r
IOError: [Errno 2] No such file or directory: '/tmp/airports/st
127.0.0.1 - - [26/Jun/2015 13:17:38] "GET /new_async?city=drs H
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /new HTTP/1.1" 200 85
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /jquery.min.js HTTP/1
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /fly_new.js HTTP/1.1"
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /flight.css HTTP/1.1"
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /ui/jquery.ui.widget.
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /ui/jquery.ui.core.js
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /ui/jquery.ui.position
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /ui/jquery.ui.autocomp
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /resources/demos/style
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /resources/demos/style
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /weiss.png HTTP/1.1" 4
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /bb.jpg HTTP/1.1" 304
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /yellow.png HTTP/1.1"
127.0.0.1 - - [26/Jun/2015 13:18:06] "GET /s.jpeg HTTP/1.1" 304
127.0.0.1 - - [26/Jun/2015 13:19:52] "GET /new_async?city=drs H
127.0.0.1 - - [26/Jun/2015 13:19:56] "GET /new_async?city=drs H
127.0.0.1 - - [26/Jun/2015 13:19:59] "GET /new_async?city=dresde
127.0.0.1 - - [26/Jun/2015 13:20:09] "GET /new_async?city=Colon
127.0.0.1 - - [26/Jun/2015 13:20:18] "GET /new_async?city=Berlin
(DEBUG:notifier:write event: /tmp/userchoice/trip/Date)
(DEBUG:notifier:write event: /tmp/userchoice/trip/Class)
127.0.0.1 - - [26/Jun/2015 13:20:34] "POST /new_date HTTP/1.1"
(DEBUG:notifier:write event: /tmp/userchoice/trip/Note)
(DEBUG:notifier:write event: /tmp/userchoice/trip/DepCity)
(DEBUG:notifier:write event: /tmp/userchoice/trip/DepCode)
(DEBUG:notifier:write event: /tmp/userchoice/trip/DestCity)
(DEBUG:notifier:write event: /tmp/userchoice/trip/DestCode)
(DEBUG:notifier:write event: /tmp/userchoice/trip/Lat)
(DEBUG:notifier:write event: /tmp/userchoice/trip/Long)
(DEBUG:notifier:write event: /tmp/userchoice/trip/Latt)
(DEBUG:notifier:write event: /tmp/userchoice/trip/Longg)
(DEBUG:notifier:write event: /tmp/userchoice/trip/User)
(DEBUG:notifier:write event: /tmp/userchoice/trip/Distance)
(DEBUG:notifier:write event: /tmp/userchoice/trip/Time)
127.0.0.1 - - [26/Jun/2015 13:20:34] "POST /new HTTP/1.1" 303 0
127.0.0.1 - - [26/Jun/2015 13:20:34] "GET /new HTTP/1.1" 200 86
127.0.0.1 - - [26/Jun/2015 13:20:34] "GET /resources/demos/style
127.0.0.1 - - [26/Jun/2015 13:20:34] "GET /resources/demos/style
127.0.0.1 - - [26/Jun/2015 13:20:34] "GET /bb.jpg HTTP/1.1" 304
```

Welcome to StealthdbFlights, Default

Change user name

Add a new flight From city:

zur Search

Erzurum

Zurich Search

Date:

Class: Select a class

Note(opt):

Save changes

Earth Moon Mars

0.1676x 0.0175x 0.0001x

Total distance is 6716.79km

Longest journey is 0km

Shortest journey is 0km

From(city)	From(ICA code)	To(city)	To(ICA code)	Distance, km	Time, min
Wisconsin Rapids	HKKR	Duesseldorf	EDDL	6716.79	530.82

06/10/2015 Economy my first flight Hide

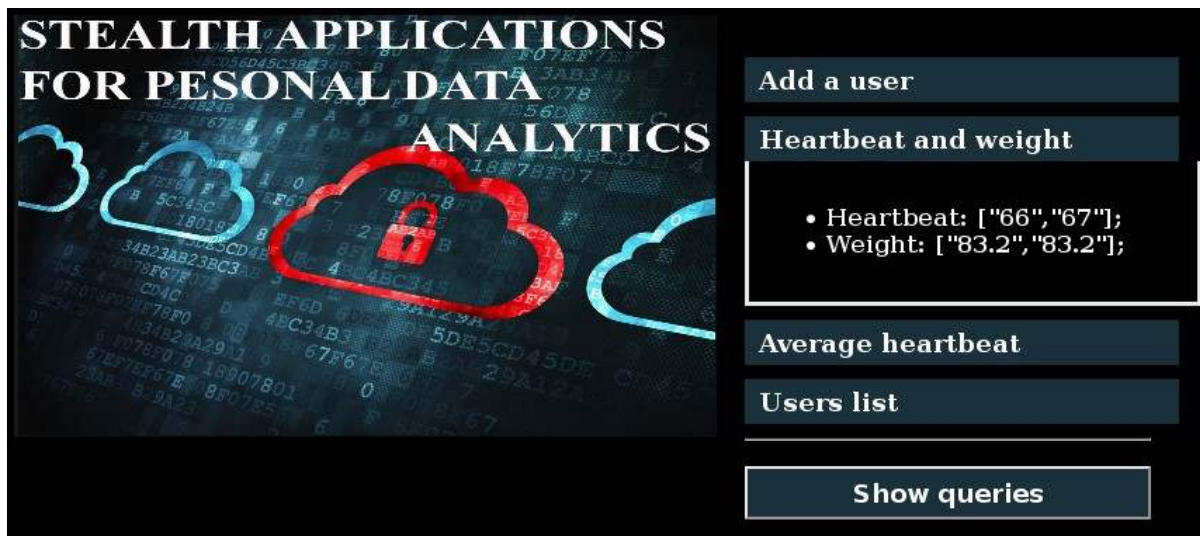
ESP Service Multiplexing

Writer:

```
USE CLOUDS 'cloud://cloud1' AND 'cloud://cloud2' WITH 'encryption,dispersion,ordered';  
DROP TABLE IF EXISTS heartsensor;  
CREATE TABLE heartsensor (frequency REAL);  
↓ INSERT INTO heartsensor (frequency) VALUES (97.4);
```

Reader:

```
USE CLOUDS 'cloud://cloud1' AND 'cloud://cloud2' WITH 'encryption,dispersion,ordered';  
MODE debug;  
SELECT AVG(frequency) FROM heartsensor WHERE frequency > 0.0 FOREVER;
```



**STEALTH APPLICATIONS
FOR PERSONAL DATA
ANALYTICS**

Add a user

Heartbeat and weight

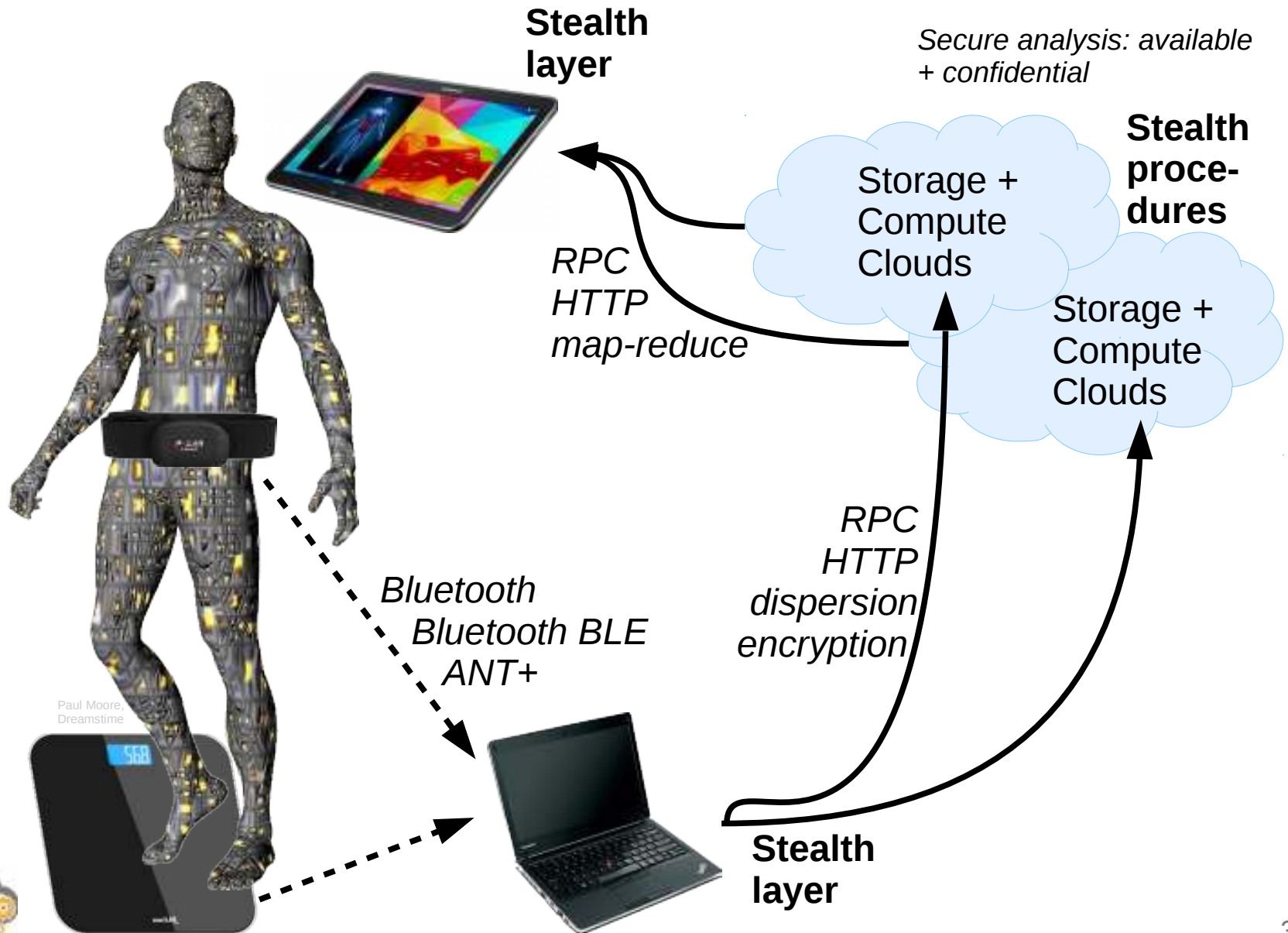
- Heartbeat: [\"66\", \"67\"];
- Weight: [\"83.2\", \"83.2\"];

Average heartbeat

Users list

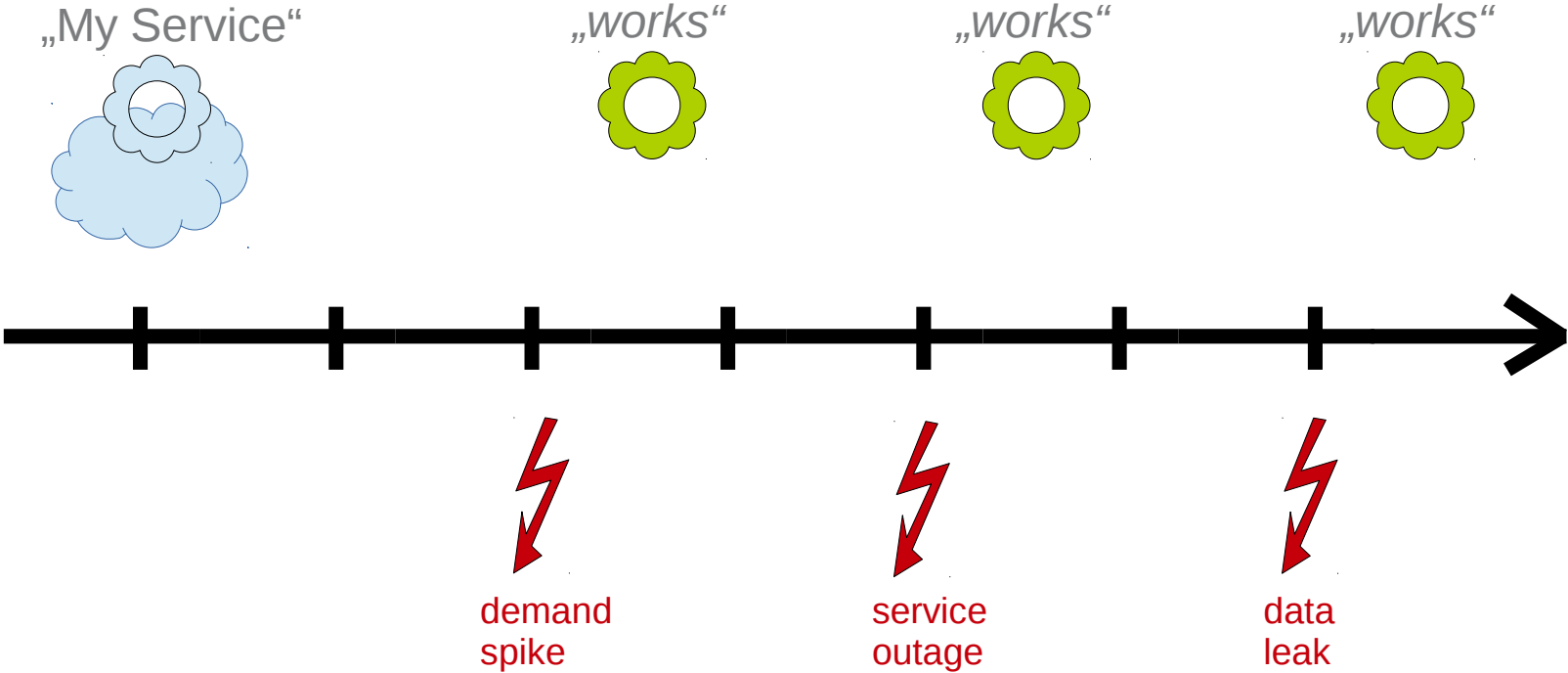
Show queries

ESP Service Multiplexing



Cloud-native & stealthy applications

Outlook: Tomorrow's cloud applications and services



Wrap-Up

New application/service models emerging for clouds:

- Cloud-Native Applications → resilience, scalability
- Stealth Applications → availability, confidentiality

SPLab to advance and transfer research + tooling in these areas!

UCC 2014/CLASP: workshop paper on stealth algorithms

NetSys 2015: demo paper on stealth queries

BlackSeaCom 2015: extended demo paper on stealth stream processing

UCC 2015: full paper on StealthDB ✓

UCC 2015/CloudAM: workshop paper on CNA evaluation

~autumn 2015: habilitation manuscript on risk minimisation in the cloud

References

- [BBT+15] Sandro Brunner, Martin Blöchlinger, Giovanni Toffetti, Josef Spillner, Thomas M. Bohnert:
Experimental Evaluation of the Cloud-Native Application Design.
4th International Workshop on Clouds and (eScience) Applications Management (CloudAM), Limassol, Cyprus, December 2015. Submitted for review.
- [SBS+15] Josef Spillner, Martin Beck, Alexander Schill, Thomas M. Bohnert:
Stealth Databases: Ensuring User-Controlled Queries in Untrusted Cloud Environments.
8th IEEE/ACM International Conference on Utility and Cloud Computing (UCC), Limassol, Cyprus, December 2015.
- [Spi15] Josef Spillner:
Secure Distributed Data Stream Analytics in Stealth Applications. (Demo)
3rd International Black Sea Conference on Communications and Networking (BlackSeaCom), Constanța, Romania, May 2015.
- [SMS15] Josef Spillner, Lorenzo Miori, Julian Sanin:
Stealth Apps for Secure Personal Data Analytics in the Cloud. (Demo)
2nd International Conference on Networked Systems (NetSys), Cottbus, Germany, March 2015.
- [SS14b] Josef Spillner, Alexander Schill:
Algorithms for Dispersed Processing.
1st International Workshop on Advances in Cloud Computing Legislation, Accountability, Security and Privacy (CLASP), London, UK, December 2014.
- [SS14a] Josef Spillner, Alexander Schill:
Towards Dispersed Cloud Computing.
2nd IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Chișinău, Moldova, May 2014.