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
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
Cloud-native & stealthy applications

Motivation: Today's cloud applications and services

„My Service“  „slow!!!“  „not working!!!“  „never again!!!“

demand spike  service outage  data leak
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

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

- **Steganography (S)**: goal: privacy
- **Dispersion (D)**: goal: availability; implied: privacy, confidentiality
- **Compression (C)**: goal: capacity
- **Encryption (E)**: Contributions; goal: confidentiality

Network:
- S → C → S+C
- S+C → D+D → S+C+D
- C → E+C → C+E+S
- S+C+D → E+S
- D+D → E+C+D
- S+C+D → D+E
- D+E → E+C+D
- E+C+D → E+S
- E+S → E+C+D
Data coding trade-offs

- Confidentiality
- High-grade encryption
- Availability
- Processable dispersed & encrypted data
- Capacity efficiency
- Replication without encryption
- Original data
- Runtime efficiency
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: 
\[ availability = \sum_{i} \binom{n}{i} a_1^i (n - a_1)^{n-i} \]

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

<table>
<thead>
<tr>
<th>Service</th>
<th>Availability</th>
<th>Capacity</th>
<th>Price</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>google-drive</td>
<td>99.90 %/a</td>
<td>unlimited/elastic free</td>
<td>1+1</td>
<td></td>
</tr>
<tr>
<td>amazon-s3</td>
<td>98.86 %/a</td>
<td>unlimited/elastic free</td>
<td>1+0</td>
<td></td>
</tr>
<tr>
<td>at&amp;t</td>
<td>99.50 %/a</td>
<td>unlimited/elastic free</td>
<td>1+1</td>
<td></td>
</tr>
<tr>
<td>linode</td>
<td>99.95 %/a</td>
<td>unlimited/elastic free</td>
<td>1+1</td>
<td></td>
</tr>
<tr>
<td>apple-icloud</td>
<td>99.65 %/a</td>
<td>unlimited/elastic free</td>
<td>1+1</td>
<td></td>
</tr>
</tbody>
</table>

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

Algorithm: PiCav
Tracing: Important
Determine distribution

Service:
- [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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time</th>
<th>Services</th>
<th>Configuration</th>
<th>Availability</th>
<th>Price</th>
<th>Capov</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>fixed[sp]</strong></td>
<td>41.49</td>
<td>T1, T10, T2, T3, T4, T5, T6, T7, T8, T9</td>
<td>{1+0, 1+0, ..., 1+0}</td>
<td>error, no solution available</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>fixed[re]</strong></td>
<td>39.86</td>
<td>T1, T10, T2, T3, T4, T5, T6, T7, T8, T9</td>
<td>{1+0, 0+1, ..., 0+1}</td>
<td>availability=1.0000 price=10.00 capov=...</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>staggered[pl]</strong></td>
<td>386.44</td>
<td>T1, T10, T2, T3, T4, T5, T6, T7, T8, T9</td>
<td>{1+x, 1+x, ..., 1+x}</td>
<td>error, solution found but rt exceeded</td>
<td></td>
<td></td>
</tr>
<tr>
<td>proportional[av/ca/pr]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>absolute[av/ca/pr]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>random</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>combinatorial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>staggered[co]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>picav</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>picav+[av/ca/pr]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$C_n \subseteq \mathcal{C} \in \mathcal{P}(\mathcal{C}) = \{\{T_1\}, \{T_1, T_2\}, ..., \{T_1, ..., T_n\}\}$$

= 16 assignment algorithm variants, 4 combinatorial x 3 staggered = 2 optm.
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

Emulation of Data Availability on Scenario 'goldstandard'

Availability (%) vs. Time (s)

- Target
- Combinatory
- Picav
- Absolute
- Picav+
- Proportional

Time scale: 0 to 2500 seconds
Stealth processing
Stealth pipeline

Compute
- Stealth Processing
  - Algorithms
  - Optimised execution
  - Programming model

Networking / Storage
- Stealth Transmission
  - Data relation schemas
  - Data transport

Stealth Distribution
- Distribution assignment
- Fragment ordering

Stealth Coding
- Combined coding
- Steganography
- Encryption
- Compression
- Recursive coding
- Forward error correction

Availability
- NP stability

Sampling

Visualisation
- Effective control

Control vertical

Techniques: Stealth search and arithmetics

Technique: Redundancy search

Relation schemas: DRS, FRS

Technique: PICav+

Technique: Bit expansion

Technique: Bit splitting

Static service selection

Policies

Service access

Operator distribution

Configuration vertical

Usage pipeline/vertical
Stealth layer

RMS: resource & service multiplexing

- Local resources
- Storage resource service
- Network resource service
- Compute resource service
- Network multiplexer/proxy
- Event stream processing
- Database service
- File storage service
- Stream storage service
- Stealth layer
  - Application
  - Application service (SaaS)
Stealth layer

Stealth layer: Coverable cloud service evolution

s: cloud services

[Image sources: dreamstime.com, bitrebels.com, suitsofarmour.com]
Showcase: StealthDB

Listing 1  Sender commands

```bash
$/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

```bash
$/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?

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

Encrypted Dispersed Processing (Stealth Processing):
- homomorphic encryption
- order-preserving encryption
- searchable encryption

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

map-carry-reduce

map-reduce

dispersed and encrypted data
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

```python
>>> 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

```sql
>>> SELECT MIN(chf) FROM cloudsalary OPTIMIZE FOR 'performance,precision';

31

(MSG: aggregate cloud:'mem://fastram')

(MSG: decryption:privkey=PrivateKey(l=276206633765143932185133659008305273600,m=7779963786372083604813961003219226636),pubkey=PublicKey(n=276206633765143932218448800529509760119,n_sq=76290104535872348142321164364999188369598776886438439468947400461280922894161,g=276206633765143932218448800529509760120),entry=7445792220222805825191130243406918283979044730019820823958984398650514208122

(MSG: decryption:decrypt-safe)

(MSG: decryption:decryption:optimized:0)

(MSG: decryption:decryption:optimized:0)

(MSG: encryption:decrypt-safe)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)

(MSG: encryption:optimized:0)
Performance (depends on protection!)

Bluetooth Heartbeat Monitoring Delays and Throughputs

Delay (s) vs. Phase and Iteration

Throughput (events/s)
File Service Multiplexing
DB Service Multiplexing
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;
```
ESP Service Multiplexing

Stealth layer

Secure analysis: available + confidential

Storage + Compute Clouds

RPC HTTP map-reduce

RPC HTTP dispersion encryption

Storage + Compute Clouds

Stealth procedures

Bluetooth BLE ANT+

Stealth layer

RPC HTTP

Paul Moore, Dreamstime
Cloud-native & stealthy applications

Outlook: Tomorrow's cloud applications and services

„My Service“ „works“ „works“ „works“
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. 

[Spi15] Josef Spillner: 
Secure Distributed Data Stream Analytics in Stealth Applications. (Demo) 

[SMS15] Josef Spillner, Lorenzo Miori, Julian Sanin: 
Stealth Apps for Secure Personal Data Analytics in the Cloud. (Demo) 

[SS14b] Josef Spillner, Alexander Schill: 
Algorithms for Dispersed Processing. 

[SS14a] Josef Spillner, Alexander Schill: 
Towards Dispersed Cloud Computing. 
2nd IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Chișinău, Moldova, May 2014.