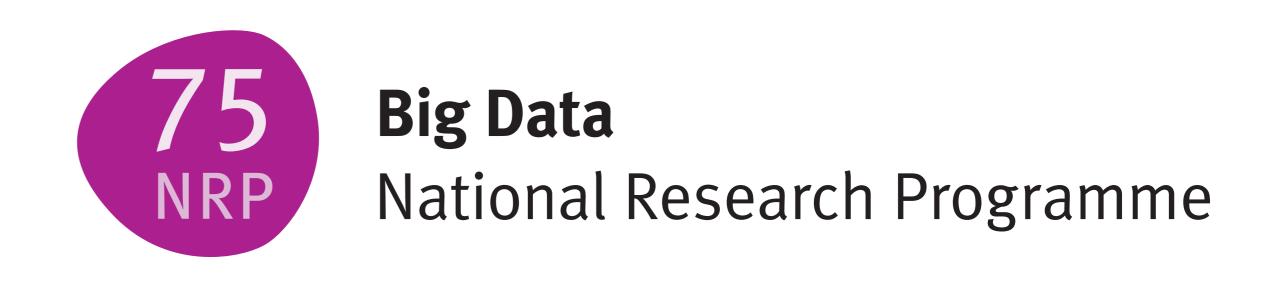# Big Data Monitoring
Dmitriy Traytel and David Basin
ETH *zürich*

The Online Monitoring Problem

event stream

verdict stream

formal policy

$\Box \forall c. \forall s. \, ssh\_login(c, s) \wedge$
$((\Diamond_{[1min,20min]} net(c)) \wedge$
$\Box_{[0,1d]}(\blacksquare_{[0,0]} net(c) \to \Diamond_{[1min,20min]} net(c))) \to$
$\Diamond_{[0,1d]} \blacklozenge_{[0,0]} ssh\_logout(c, s)$

monitor

explanation

translation (manual)

SSH sessions must not last longer than 24h.

informal policy

Grand Challenge

scalability

high volume
GB TB PB …
high velocity
10^3 10^6 10^9 …
events per second

aggregation over data
for client X number of logins <3

quantification over data
for all clients there exist id

qualitative temporal constraints
before, after

quantitative temporal constraints
within 1 hour

tradeoff

logical connectives
and, or, not

out-of-order
or
lost events

expressiveness of the specification language

imprecision of data

Methodology

1 split data

30.08.16 12:12:21 *ssh_login(Dmitriy, ServerETH)*
30.08.16 12:13:35 *ssh_login(Christian, ServerSNF)*
30.08.16 12:20:03 *net(Christian)*
30.08.16 12:21:53 *net(Dmitriy)*, *net(Christian)*
30.08.16 12:33:35 *ssh_logout(Christian, ServerSNF)*
…
31.08.16 12:10:18 *net(Dmitriy)*
31.08.16 12:16:42 *net(Dmitriy)*
31.08.16 12:20:15 *ssh_logout(Dmitriy, ServerETH)*

challenges:
- preserve violations
- minimize data duplication
- minimize communication between monitors for individual slices (ideally: no communication)
- do all of this online

2 parallelize

use sequential monitors as black boxes

fault tolerance via Big Data frameworks

Spark
Flink

3 validate

synthetic case studies

realistic case studies

Zürcher Kantonalbank   NEC
Wireless Registry

existing tool
MONPOLY
language: $MFOTL_\Omega$

practice

theory

new tool
AERIAL
language: MDL

1 one monitor
different languages

LTL LTL_{FO}
MTL   MFOTL
MDL $MFOTL_\Omega$

motto
pay only for the features in use

2 scalably monitorable
language fragments

¬ ∃ Σ

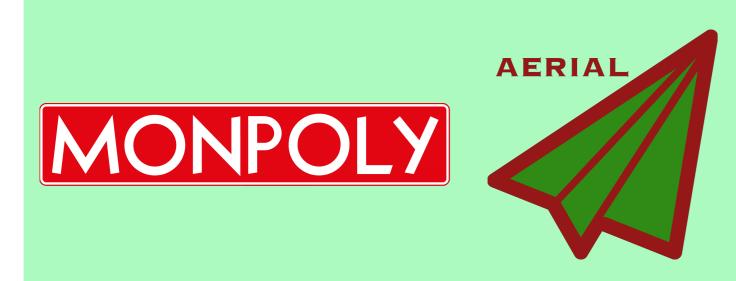**3** monitor imprecise data

Project's Aims

1 Develop, implement, and validate **scalable** and **efficient** algorithms that solve the Online Monitoring Problem

2 Delimit **expressive** specification languages that cater for scalable and efficient monitoring algorithms

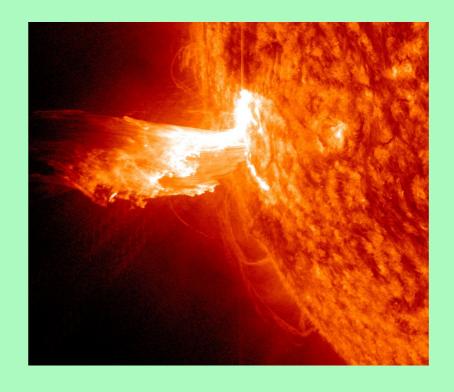**3** Extend monitoring algorithms to account for **imprecise** data

Expected Results

MONPOLY   **AERIAL**

scalable versions of our monitoring tools

characterize which policies can be monitored on Big Data

↓ enable

predictable and trustworthy "querying" of Big Data by monitoring based on an expressive specification language with a solid logical foundation
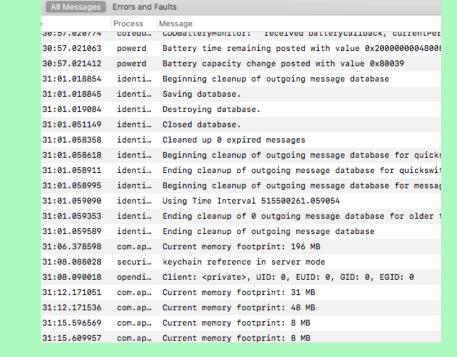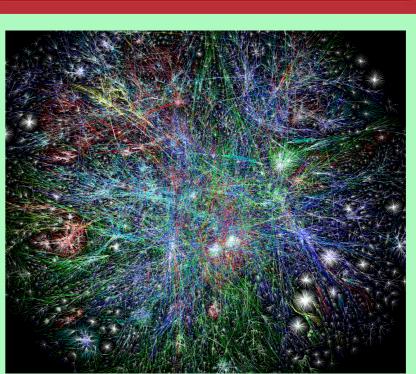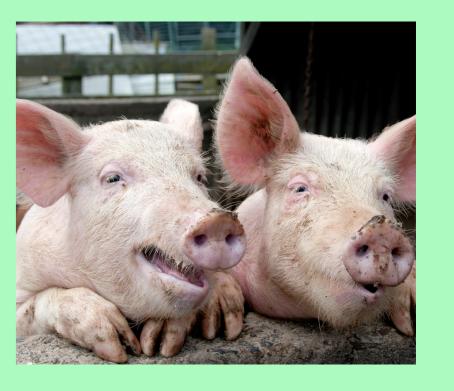
Collaborations with Other Projects

I WANT YOUR DATA*

*and your policies/properties/conjectures