Observing Internet Path Transparency

Brian Trammell, ETH Zürich

(with Mirja Kühlewind, Elio Gubser, Piet De Vaere,
Iain Learmonth, Gorry Fairhurst,
Roman Muntener, and Stephan Neuhaus)

AIMS 2017, CAIDA, San Diego, 1 March 2017

mami
measurement and architecture for a middleboxed internet

measurement   **architecture**   **experimentation**
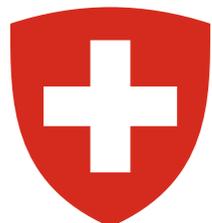
Observing Path Transparency: What and Why?

- Determine the extent to which transport-layer protocols and features are impaired by accidental and purposeful manipulation in the present Internet

  - Provide guidance for protocol engineering: which features need a fallback, which can we let fail, which will never work?

- Take simple active measurements over many *paths*, infer *conditions*, compose in space in time
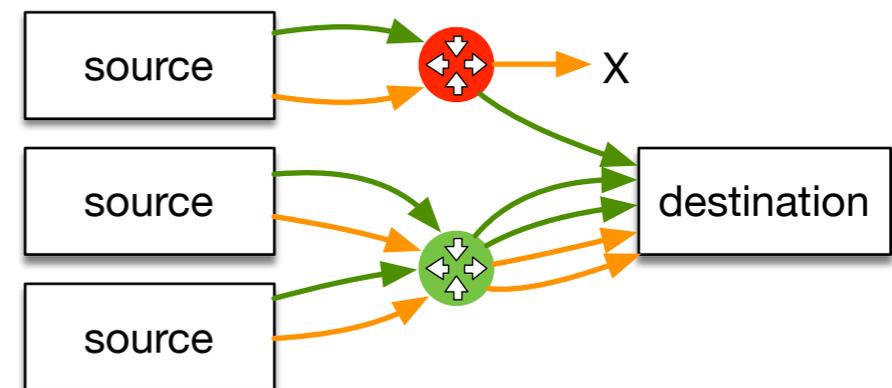
Active Measurement: Pathspider

- Tool[1] for one-sided measurement of many targets from a single source, with simultaneous passive observation of generated packets

- Plugins for ECN, TFO, DSCP

  - Extension support

  - Connectivity dependency

- Automation[2] of cloud-originated measurement of public targets[3]

  - Multiple-source measurement for path-dependency inference

[1] https://pathspider.net/
[2] https://github.com/mami-project/autospider-salt
[3] https://github.com/mami-project/targets
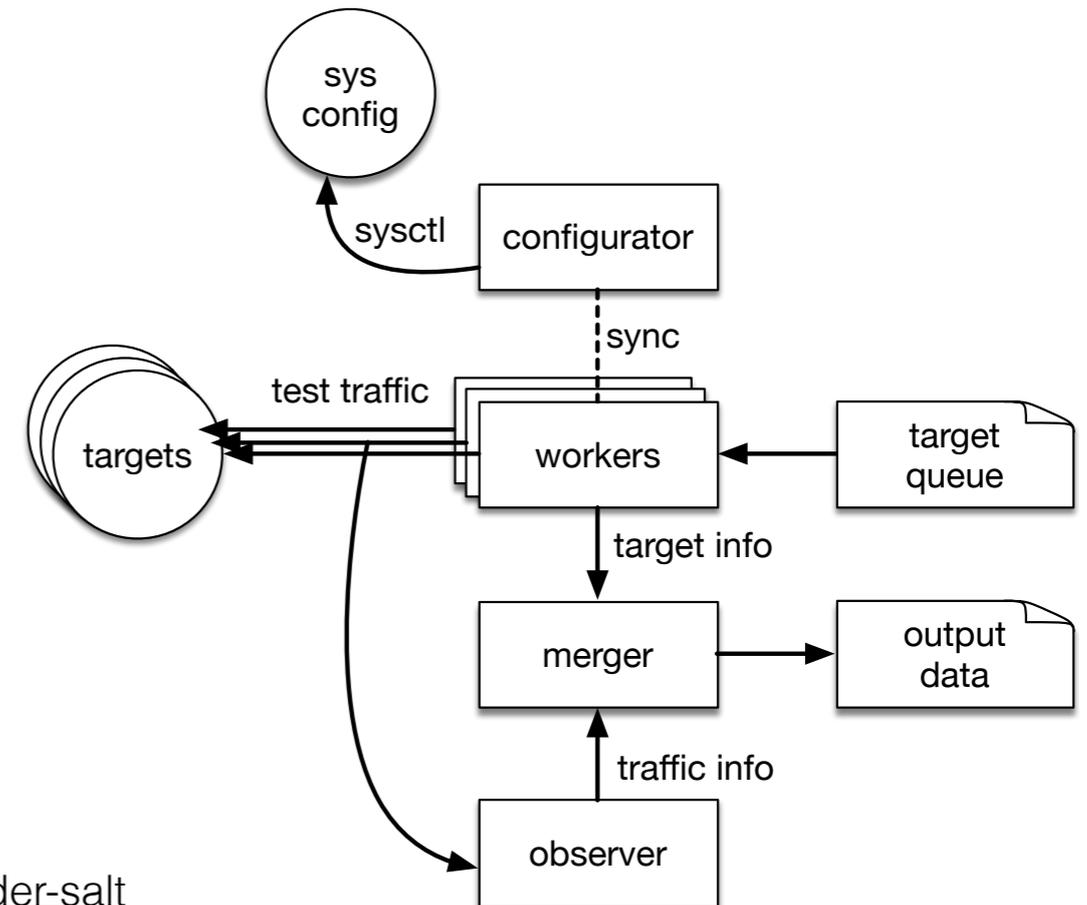
Composition and Analysis: Path Transparency Observatory

- Collect observation data as raw output from various tools (including Pathspider)

- 1st stage (raw) analysis converts these to *base* observation four-tuples:

$$\{t, p, c, v\}$$

  - t: time interval during which observation is valid

  - p: path designator, a sequence of path elements from observation point or source to target or destination

  - c: condition observed (within a defined space of conditions)

  - v: value associated with condition observed

- n^{th} stage derives *composed* observations from base observations

Design Goals:
Path Transparency Observatory

- Provide *comparability, reduction,* and *visibility* to data from different sources through a common schema for path transparency information.

- Ensure *repeatability* by providing provenance, link observations to intermediate and raw data as well as analysis code (by commit reference).

- Provide *safety* for collected data via:

  - Variable-precision, anonymizable path designators.

    - IP, prefix, AS, pseudonym-level.

  - Code reviews of contributed analyzers.

  - Human review of first stage results.

- Provide *accessibility* with a web front-end for issuing queries as well as "canned" queries for common conditions.

Design: Path Transparency Observatory
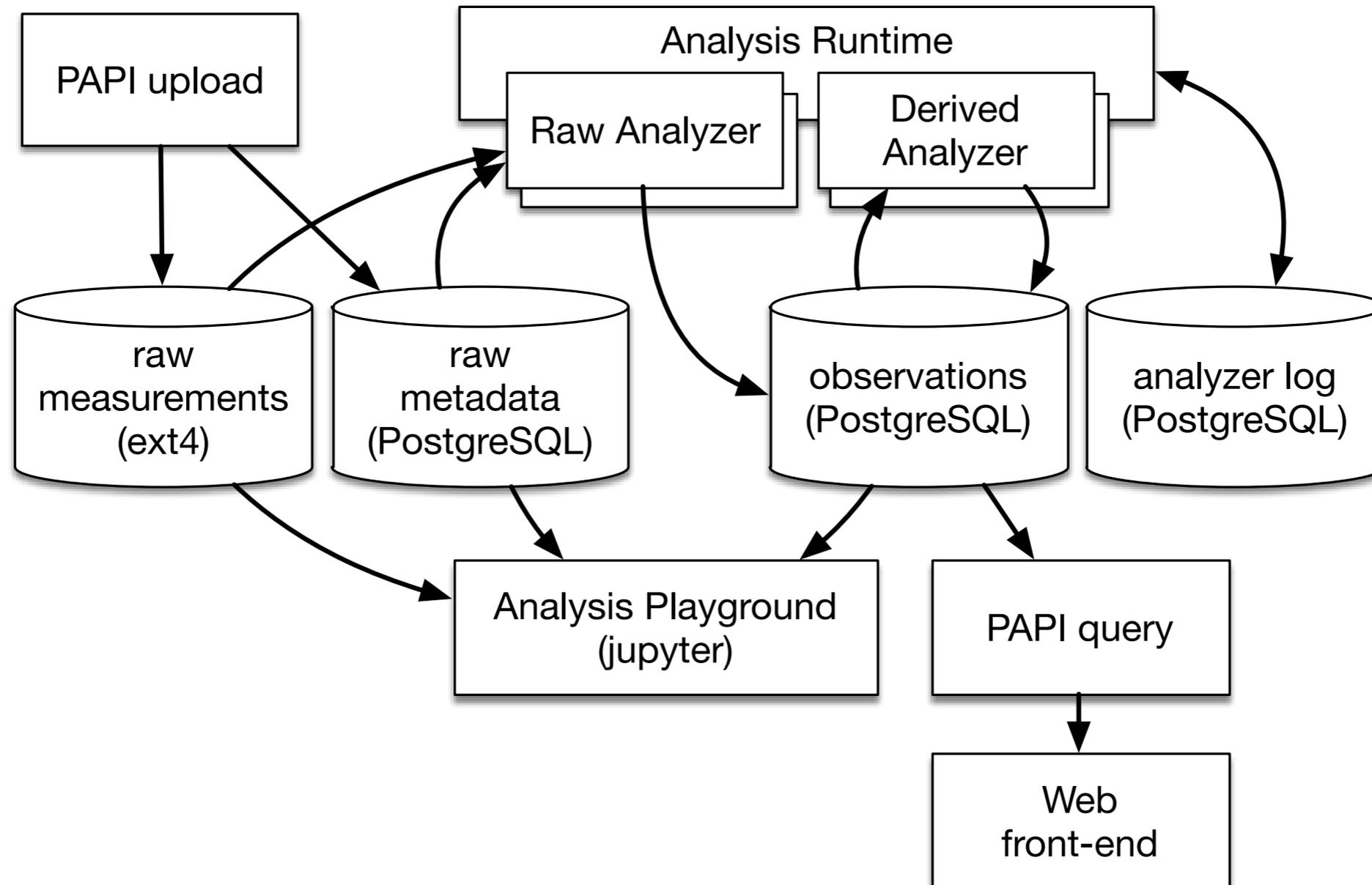
Lessons Learned:
Medium Data Suffices

- Initial design: Big Data™ compliant

  - HDFS for raw data files, Spark for raw analysis, MongoDB for observation and metadata storage, provenance *per observation*.

  - Lots of overhead for not much win

  - Rigid workflow poorly matched to research

- Reimplementation: keep it simple (and party like it's 1999)

  - Raw data in ext4, raw analysers over streams

  - PostgreSQL for observations and metadata w/ provenance and derived analysis *per observation set*.

  - Human intervention in analysis (required for review anyway).

Lessons Learned: path opacity not so different from censorship/non-neutrality

- Measurement of path-dependent ECN connectivity dependence: inferred middlebox interference far from the endpoint.

- Automated measurement reduces the noise floor, eliminates transient failure.

- What we see: failures much more likely in countries with documented heterogeneous, TCP-interfering censorship.
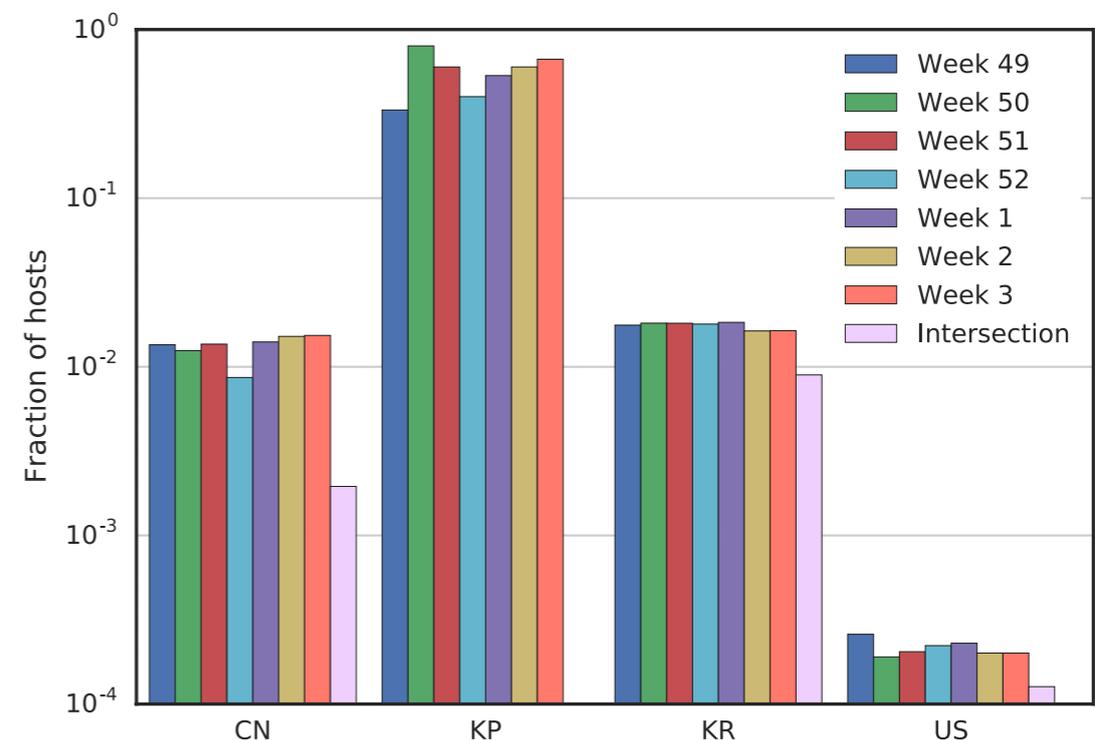
The Future

- Convergence with censorship/neutrality measurement

  - Definition of condition set in terms of OONI test specifications; integration of Pathspider with OONI.

- Transition to access network/mobile measurement

- Pay more attention to the path

  - Now we just look at endpoints, i.e. [src, *, dst]

  - Add resolution-time AS and traceroute to Pathspider

  - Explore graph databases for comparison/analysis