LazyTAP:
On-Demand Data Minimization for Trigger-Action Applications

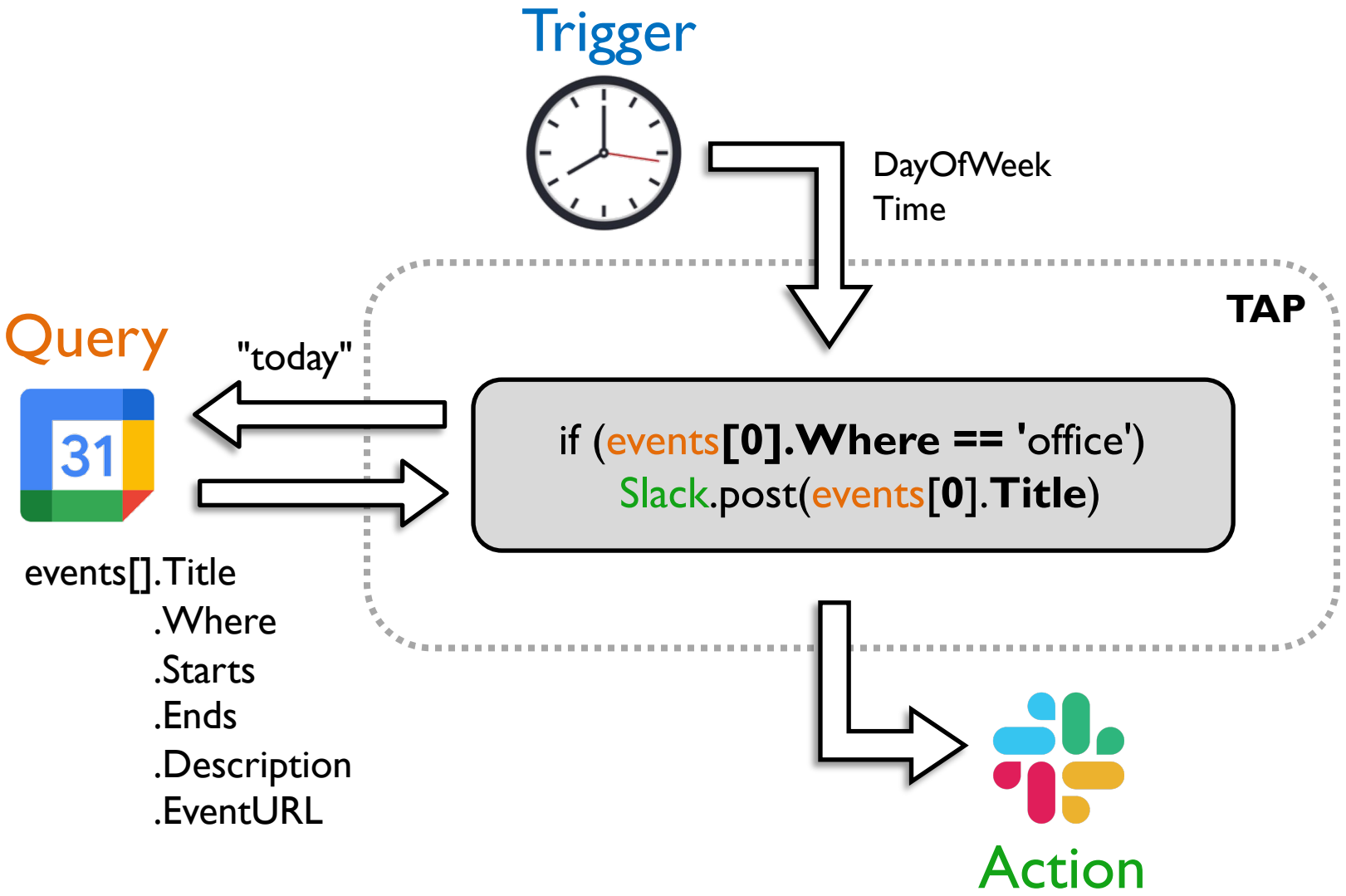
Mohammad M. Ahmadpanah Daniel Hedin Andrei Sabelfeld

IEEE Computer Security Foundations Symposium (CSF)
July 2023

Trigger-Action Platform (TAP)

- Connecting otherwise unconnected services and devices
- **Trigger** event comes, app performs an **Action**
- Additional data source with **Queries**
  - Private data e.g., calendar events, watched movies, and locations

Push-all approach in TAPs



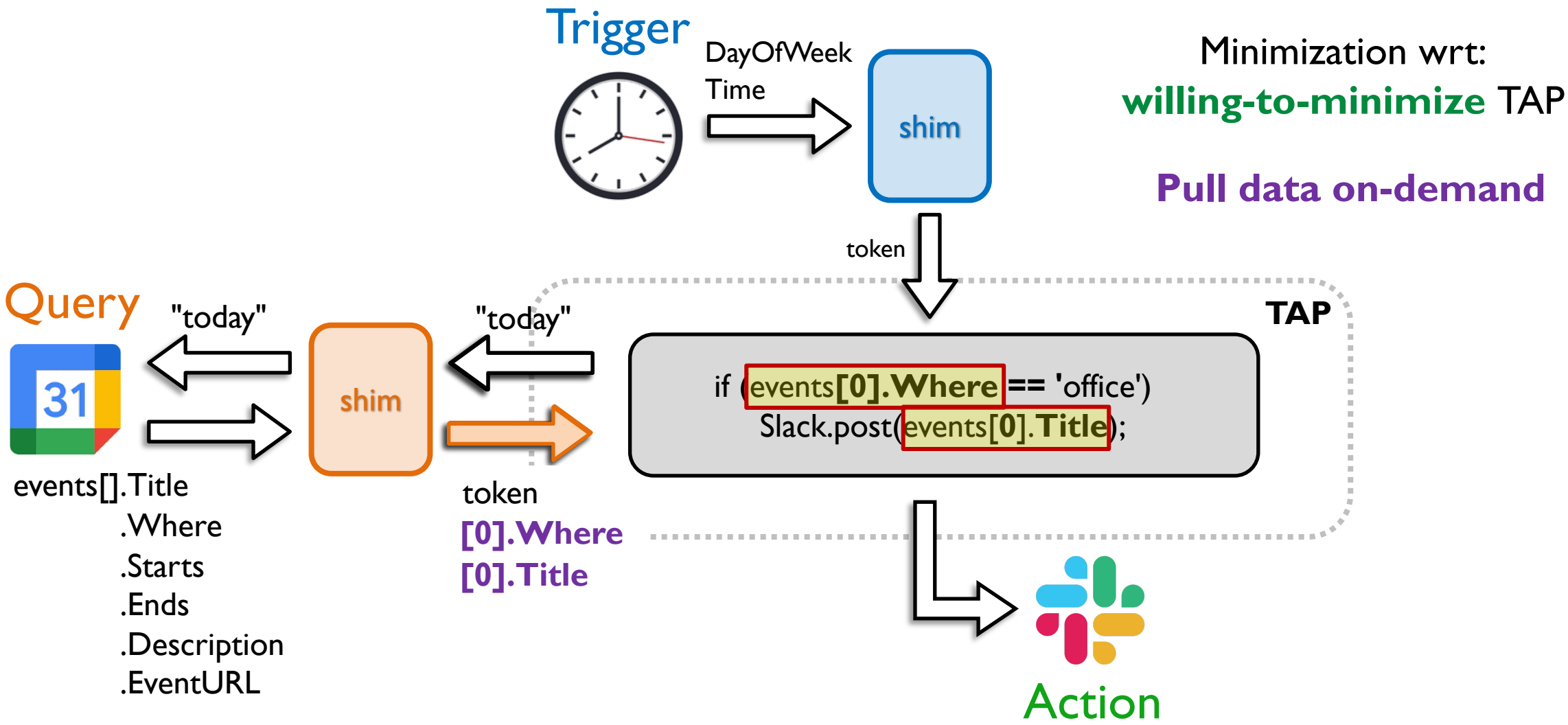
"Every morning, post the title of the first office meeting to Slack"

Push-all approach

All trigger/query data to TAP independent of the app code

at odds with ***data minimization***

LazyTAP: data minimization by design

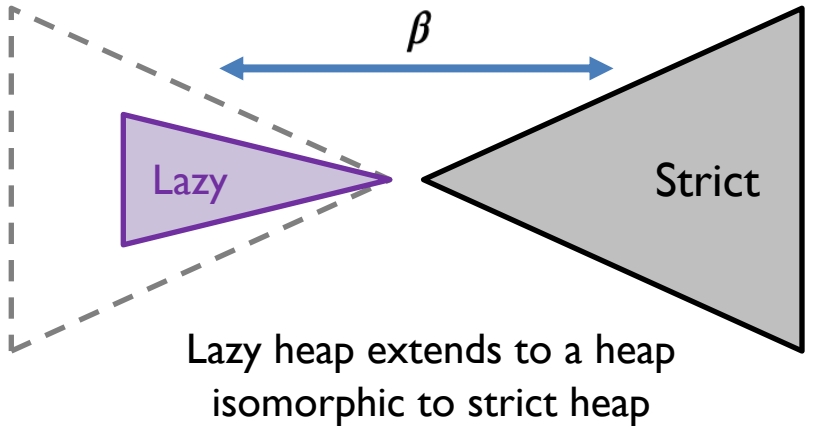


Modeling

- Modeling ***lazy runtime*** for apps
  - Remote objects, lazy query, and deferred computation by thunking
- Core language: While language with objects

$$e ::= v \mid x \mid e \oplus e \mid f(e) \mid e[e] \mid \{\} \mid T \mid Q(k, e) \mid A(m) \mid () \Rightarrow e$$

Theorem: LazyTAP is correct and at least as precise as preprocessing minimization

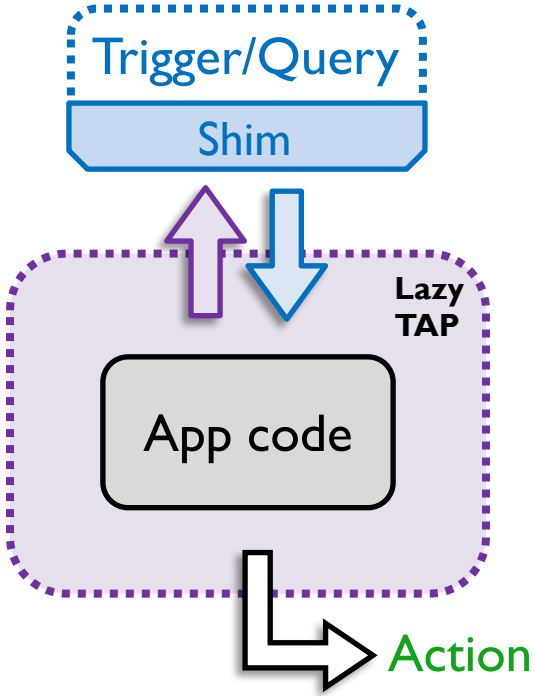


Lazy heap extends to a heap isomorphic to strict heap

Be *lazy*, be *minimized!*

On-demand minimization by design:

- **Input-sensitive** and fine-grained
- Supporting **queries** and **nondeterminism**
- **Seamless** for app developers
- **Correctness** and **precision** formally proved
- Benchmarking:
  95% over IFTTT, **38%** over static minTAP



Lazy runtime by:

- Proxied **remote objects**
- Deferred computation by **thunking**

https://www.cse.chalmers.se/research/group/security/lazytap