Decentralized Action Integrity for Trigger-Action IoT Platforms

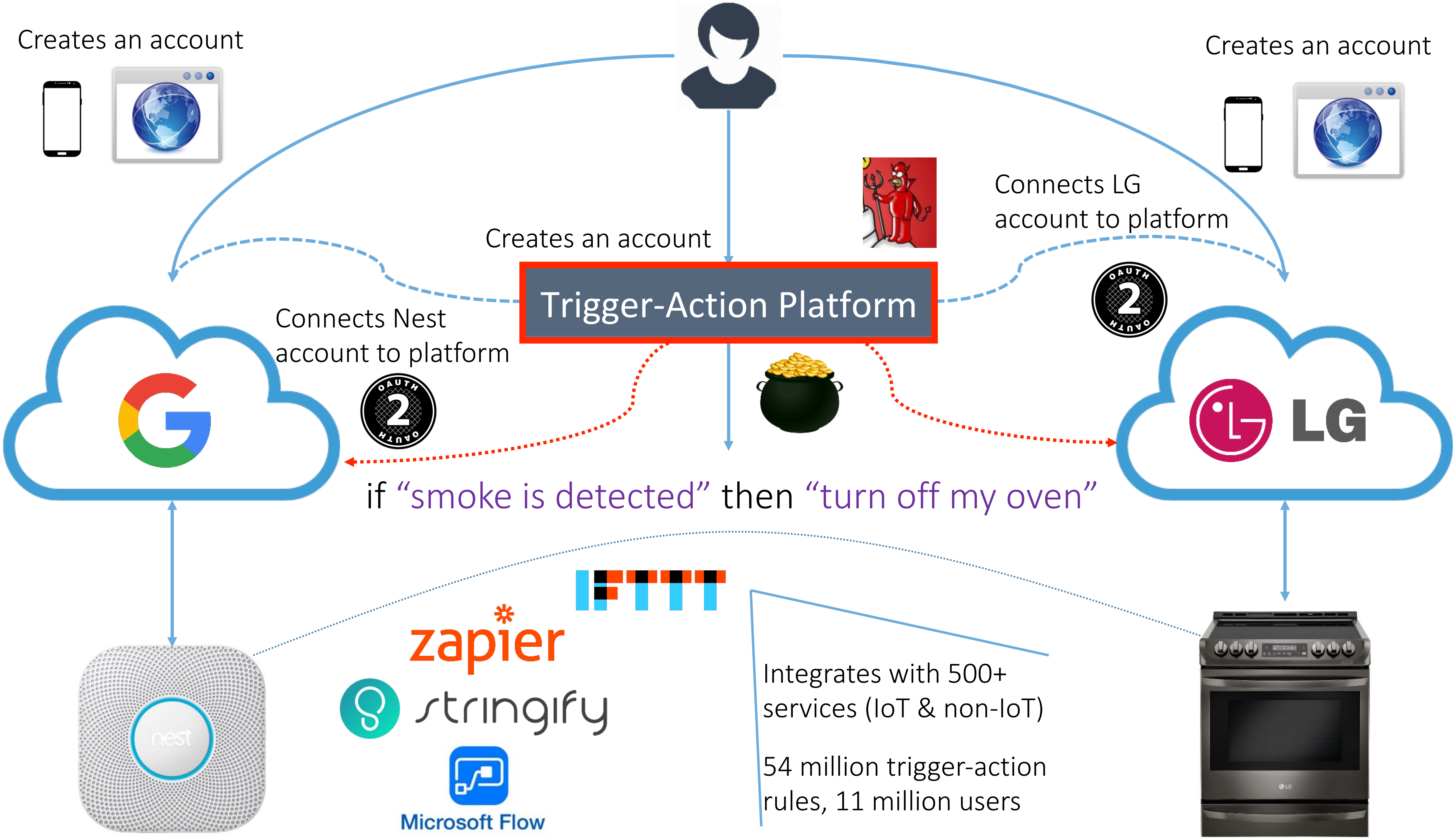
Earlence Fernandes, Amir Rahmati, Jaeyeon Jung, Atul Prakash

UNIVERSITY of WASHINGTON

UNIVERSITY OF MICHIGAN

SAMSUNG RESEARCH AMERICA

Stony Brook University



Creates an account

Creates an account

Creates an account

Trigger-Action Platform

Connects Nest account to platform

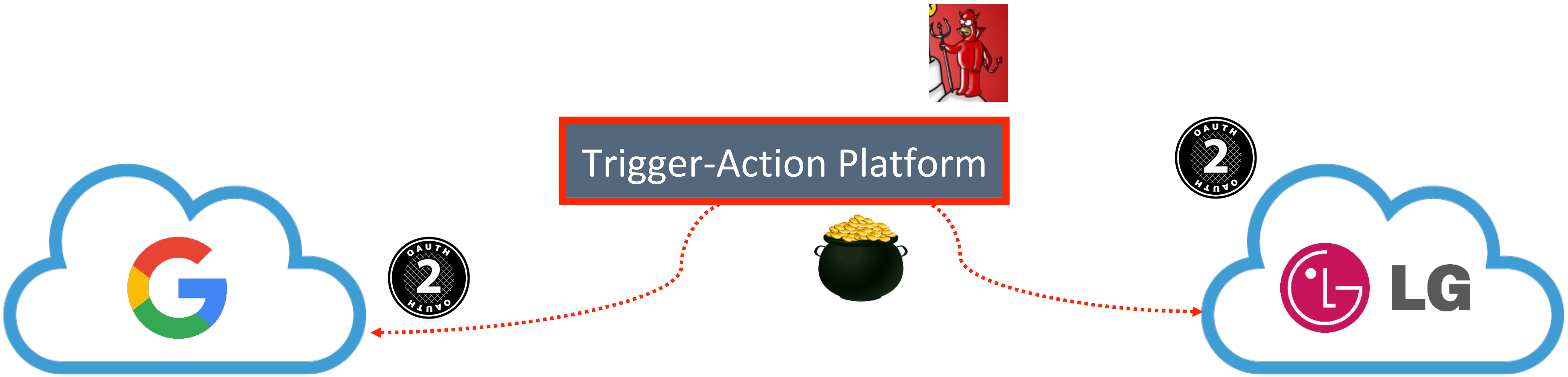
Connects LG account to platform

if “smoke is detected” then “turn off my oven”

Integrates with 500+ services (IoT & non-IoT)

54 million trigger-action rules, 11 million users

If IFTTT is Compromised, Then…

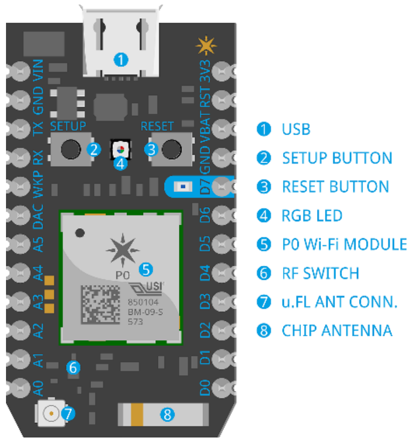


- Attackers can steal OAuth tokens to execute actions at will, independently of user rules
- If those OAuth tokens are overprivileged, the threat is made worse
  - We studied popular channels (IoT and non-IoT), and found instances of overprivilege

With Overprivileged OAuth Tokens, Attackers Can…

- Reprogram Particle Chips with Custom Firmware

  https://api.particle.io/v1/devices/device-id



- Delete Files on Google Drive

  https://www.googleapis.com/drive/v3/files/file-id

- Turn Devices On/Off Arbitrarily in a Connected Home

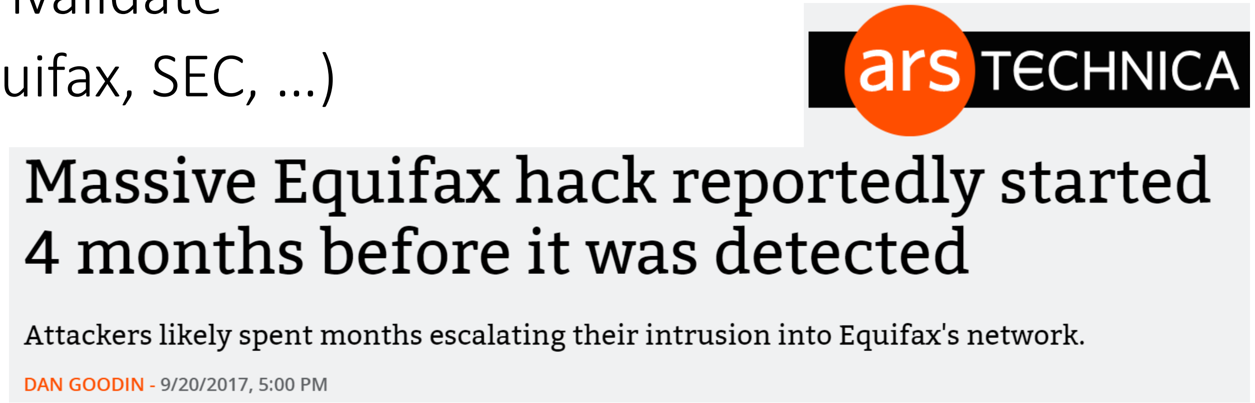
  https://api.myfox.me:443/v2/site/site-id/device/dev-id/socket/on or /off

These operations aren’t available as triggers or actions

How can we guarantee that
actions are executed according to user rules
in an untrusted trigger-action platform?

Could We Try…

- Short-lived OAuth tokens?
  - Token lifetime is very small, requiring many refresh calls
  - Upon compromise, immediately invalidate
  - BUT, detection is never timely (Equifax, SEC, …)



Massive Equifax hack reportedly started 4 months before it was detected

Attackers likely spent months escalating their intrusion into Equifax's network.

DAN GOODIN - 9/20/2017, 5:00 PM

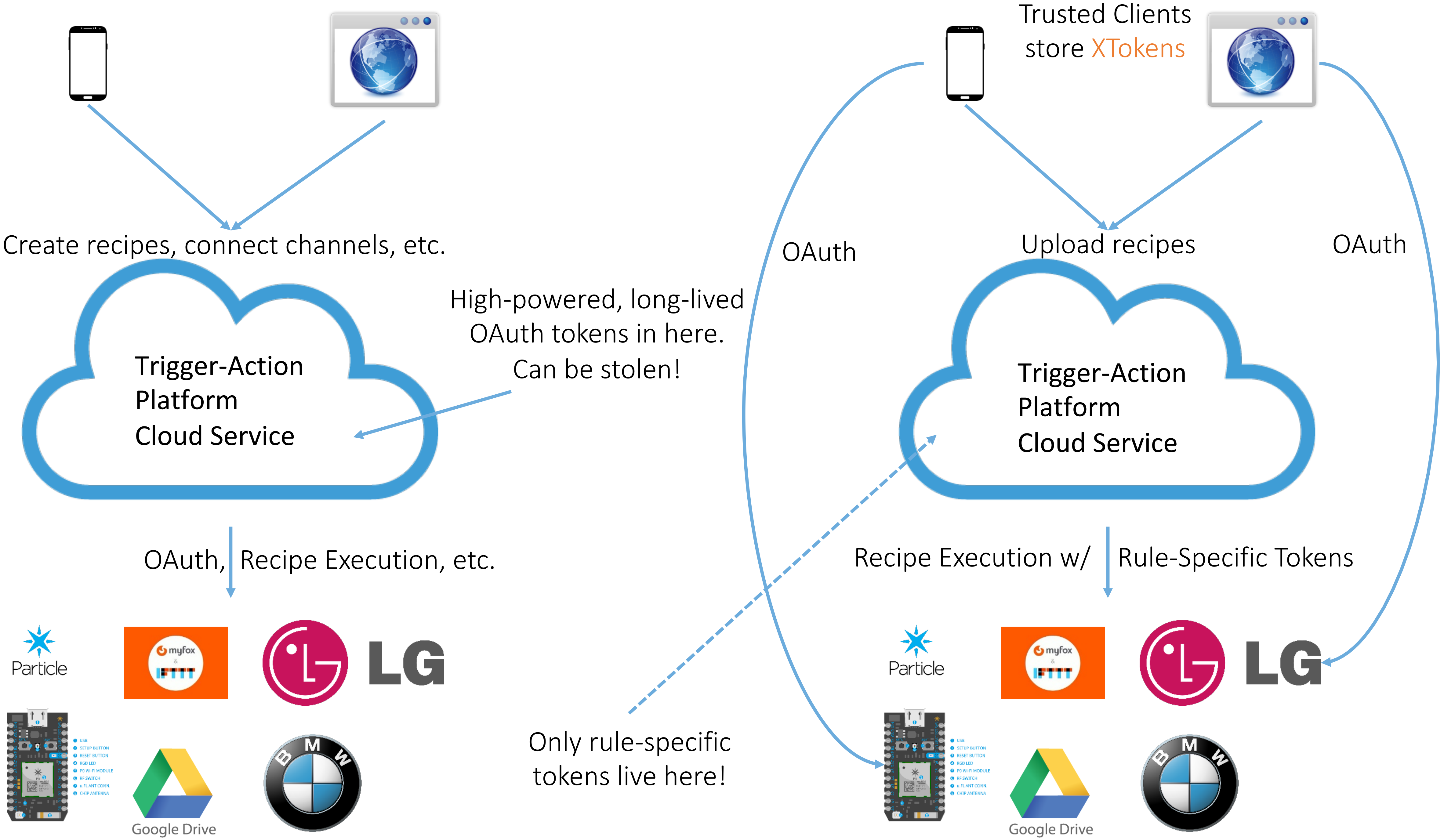
- Rule Analytics/Anomaly Det?
  - After-the-fact, damage is done
  - Does not address root cause

- Fully Decentralized Platform?
  - No high-availability, reliability

- Finely-Grained Tokens?
  - Usability problems

Challenges	Solutions
Finely-grained tokens • E.g., token only for oven.off() • Problem: attackers can still misuse	

Challenges	***Decentralized Action Integrity***
Finely-grained tokens • E.g., token only for oven.off() • Problem: attackers can still misuse	Verifiable Triggers => Rule Specific Tokens • E.g., can invoke oven.off ONLY IF holder of token can prove that trigger occurred
Trigger-action platform is untrusted • Cannot depend on it to do verification	Modified workflow: Trusted clients setup rules, Online services do verification
Usability is hurt with fine-grained tokens	XToken (transfer token): mint a rule-specific token non-interactively
Untrusted trigger-action platform can modify data as it passes through	Integrity guarantees with signatures



Create recipes, connect channels, etc.

Trigger-Action Platform Cloud Service

High-powered, long-lived OAuth tokens in here. Can be stolen!

OAuth, Recipe Execution, etc.

Trusted Clients store XTokens

Upload recipes

OAuth

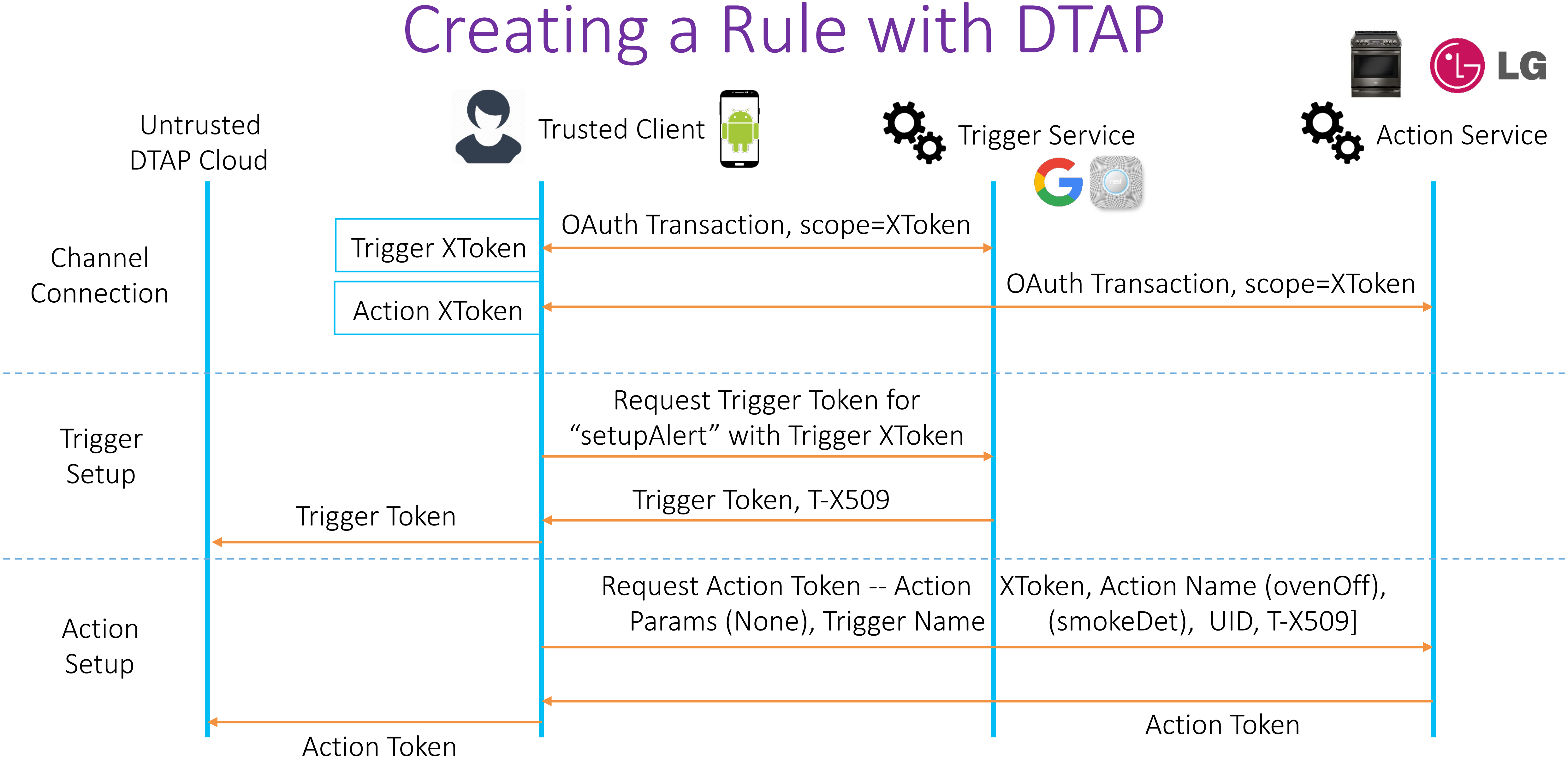
OAuth

Trigger-Action Platform Cloud Service

Recipe Execution w/ Rule-Specific Tokens

Only rule-specific tokens live here!

Creating a Rule with DTAP



Untrusted DTAP Cloud

Trusted Client

Trigger Service

Action Service

Channel Connection

Trigger XToken

Action XToken

OAuth Transaction, scope=XToken

OAuth Transaction, scope=XToken

Trigger Setup

Request Trigger Token for “setupAlert” with Trigger XToken

Trigger Token, T-X509

Trigger Token

Action Setup

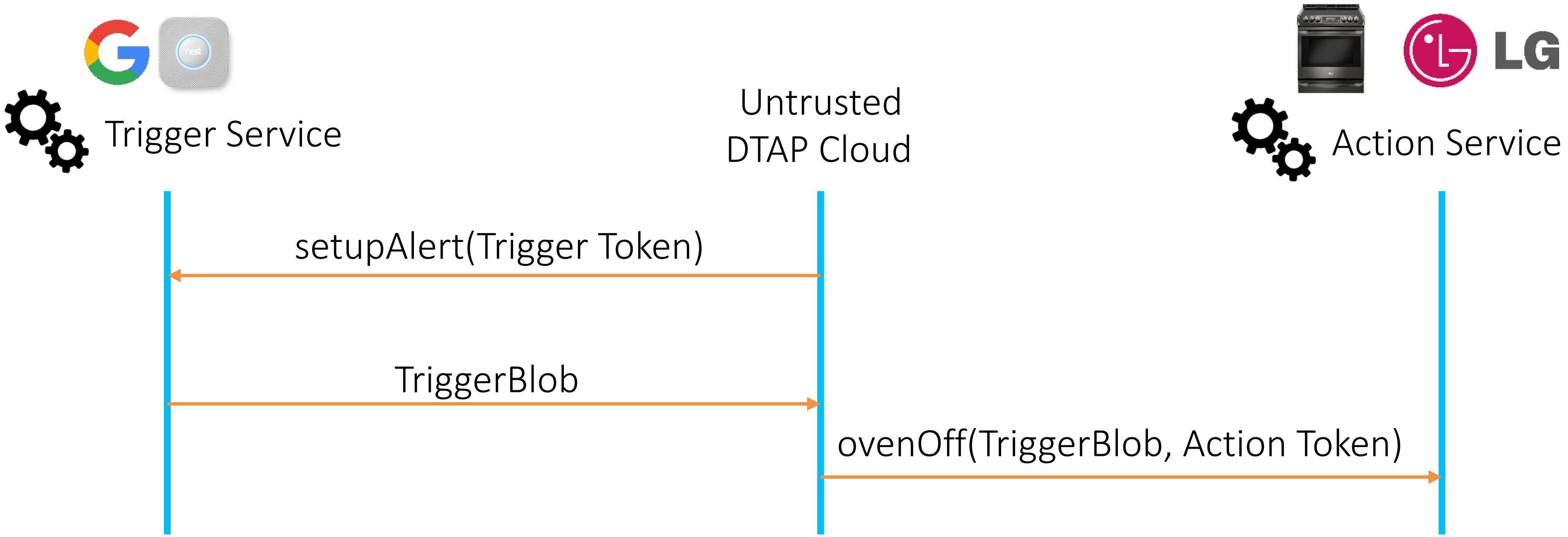
Request Action Token -- Action Params (None), Trigger Name

XToken, Action Name (ovenOff), (smokeDet), UID, T-X509]

Action Token

Action Token

Invoking Actions Requires Proof of Trigger Occurrence



Trigger Service

Untrusted DTAP Cloud

Action Service

setupAlert(Trigger Token)

TriggerBlob

ovenOff(TriggerBlob, Action Token)

Time	TTL	Trigger Name	Trigger Data	UID
12:53:34 UTC	60s	smokeDet	CO = 200ppm	ABC123

Signed using Trigger Service Private Key

Verification Procedure

Time	TTL	Trigger Name	Trigger Data	UID
12:53:34 UTC	60s	smokeDet	CO = 200ppm	ABC123

Signed using Trigger Service Private Key

- Ensure that the passed ActionToken exists
- Verify signature on trigger blob
  - Ensure Time stamp has increased
  - Verify TTL is valid
  - Check that TriggerBlob.TriggerName == ActionToken.TriggerName
  - Verify that the UID is for the current user
- Verify that the API call being made by DTAP cloud is the same as that during ActionToken creation
- Verify that function parameters match those that the trusted client gave to the action service during rule setup

Performance Evaluation

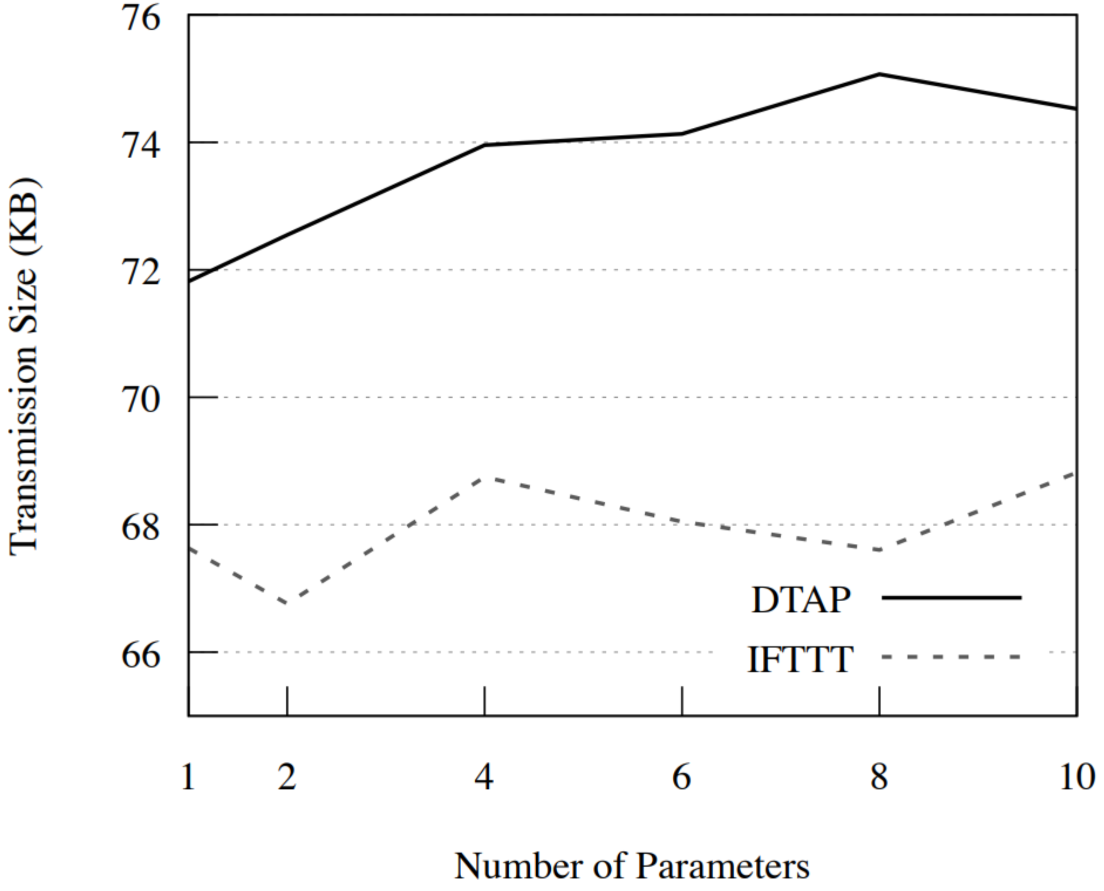
Implemented as drop-in OAuth library

Setup

If new_item == ‘buy soap’ is added to MyToDoList
Then send_email(new_item)

- Representative of a typical trigger-action rule
- Contains a condition on trigger data
- Contains transfer of data from trigger service to action service

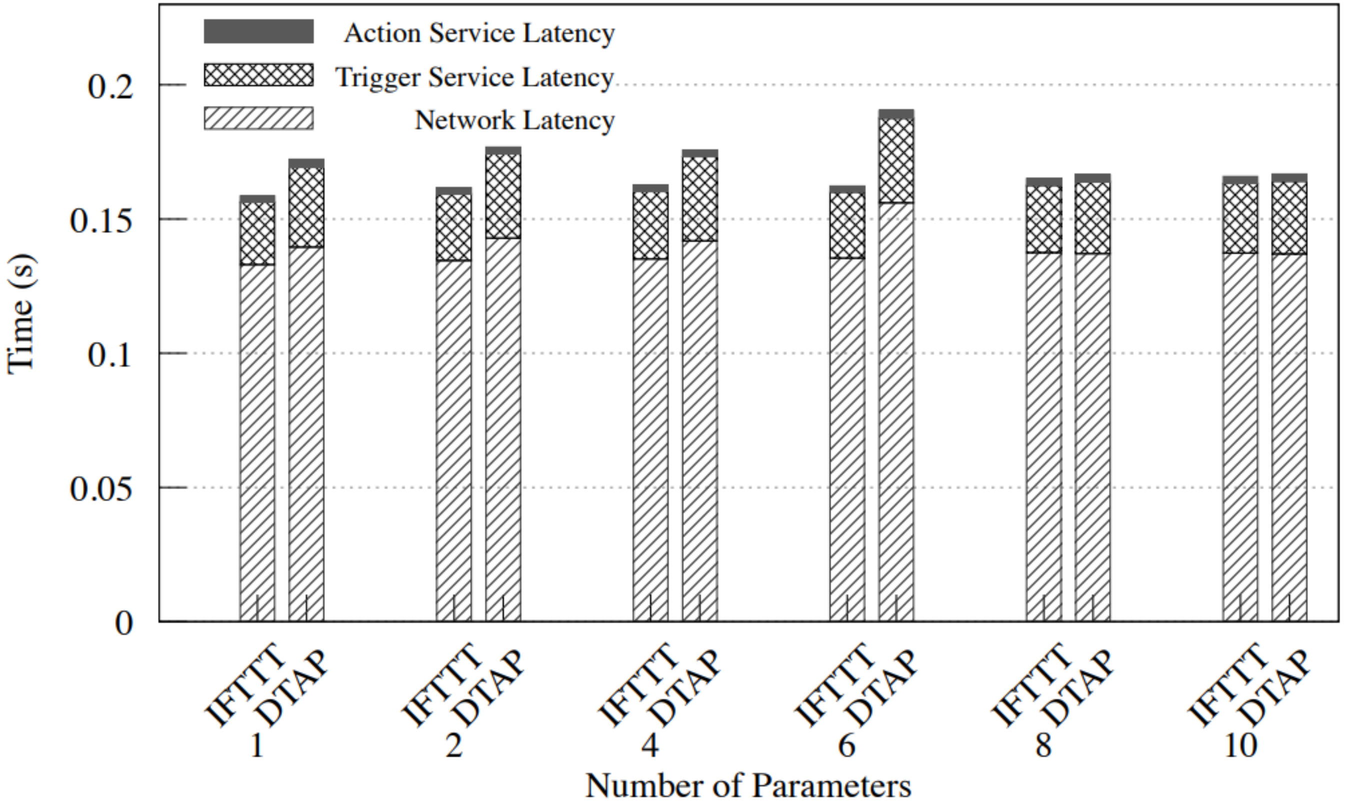
Transmission Overhead



Performance Evaluation

If new_item == ‘buy soap’ is added to MyToDoList Then send_email(new_item)

End-to-End Latency



10,000 Trigger Activations with upto 2000 concurrent requests using ApacheBench

Throughput	DTAP	IFTTT
Requests per second	94.03 (SD=8.48)	96.46 (SD=5.74)

Summary

- Emerging trigger-action platforms support stitching together various online services, including cyber-physical devices
    - BUT, if they are compromised (as is common with web apps), attackers can misuse OAuth tokens for a large number of users
- We introduced Decentralized Action Integrity
    - Rule-specific OAuth tokens with decentralized verifiable triggers
    - Uses the XTOKEN, a way to gain the power of fine-grained tokens without losing the usability benefits of coarse-grained tokens
    - Minimal performance impact & backwards-compatible with OAuth
- Clean-slate trigger-action platform design with strong integrity guarantees; first step towards removing trust from the cloud component for IoT

Decentralized Action Integrity for Trigger-Action IoT Platforms

- Emerging trigger-action platforms support stitching together various online services, including cyber-physical devices
  - BUT, if they are compromised (as is common with web apps), attackers can misuse OAuth tokens for a large number of users
- We introduced Decentralized Action Integrity
  - Rule-specific OAuth tokens with decentralized verifiable triggers
  - Uses the XTOKEN, a way to gain the power of fine-grained tokens without losing the usability benefits of coarse-grained tokens
  - Minimal performance impact & backwards-compatible with OAuth
- Clean-slate trigger-action platform design with strong integrity guarantees; first step towards removing trust from the cloud component for IoT

Earlence Fernandes earlence@cs.washington.edu

UNIVERSITY of WASHINGTON
UNIVERSITY OF MICHIGAN
SAMSUNG RESEARCH AMERICA
Stony Brook University

Macaroons

- Our work introduces decentralized action integrity as a principle
  - Our protocol is one way to enforce this principle
  - It is backwards-compatible with Oauth
  - Does not require code changes in the cloud service of TAP
- Macaroon third-party caveats + discharges can be used to implement decentralized action integrity --- but this still requires our decentralized platform architecture with Trusted Clients!
- Macaroons require a domain-specific language to implement caveats
  - For a trigger-action platform setting, this would require a different DSL for every service, because when discharging a macaroon for a third-party caveat (to obtain a verifiable trigger), each predicate is specific to the third-party online service. DTAP does not have this requirement, and is independent of the semantics of the online service APIs

Why should you trust the client?

- Developer (client) != Developer (trigger-action platform)
  - E.g., SSH, FTP, Telnet

- Few good apps emerge in app market models
  - E.g., JuiceSSH, etc.

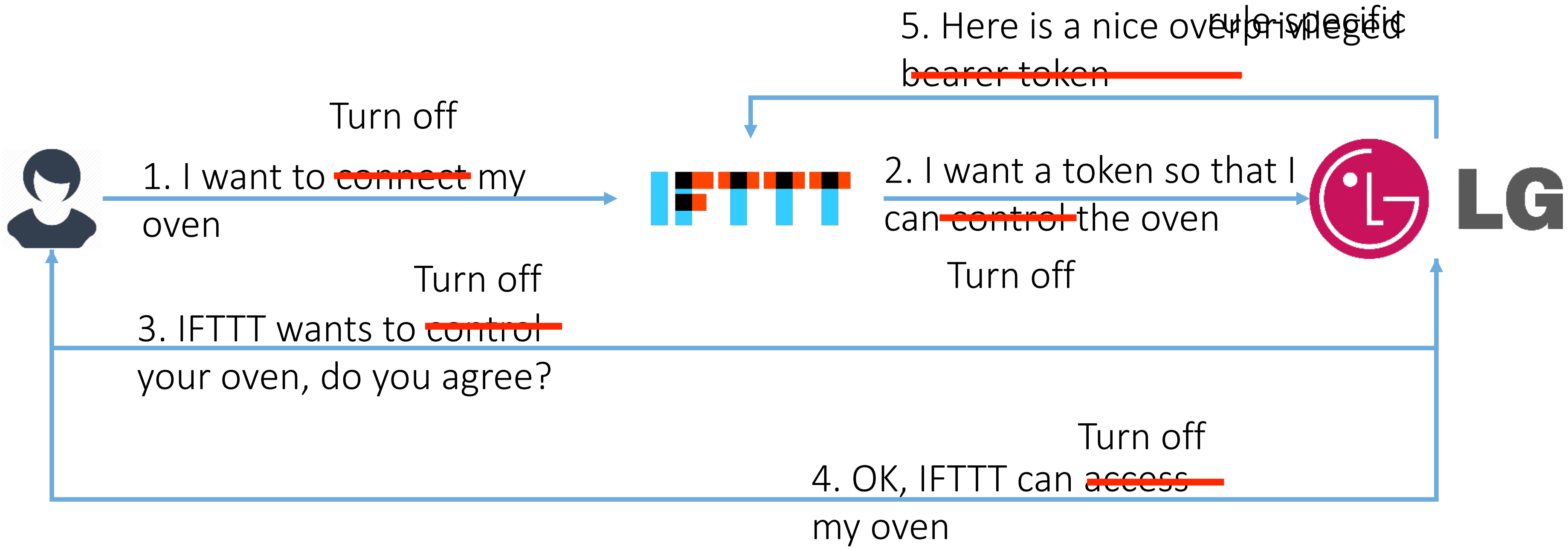
- DTAP protocol is open; designed to be implemented by anyone

- Trigger-action platform cloud service provides rule execution at scale

Finely-Grained Tokens Can Hurt Usability

ONCE per channel

ONCE per new recipe



1. I want to ~~connect~~ Turn off my oven

2. I want a token so that I can ~~control~~ Turn off the oven

3. IFTTT wants to ~~control~~ Turn off your oven, do you agree?

4. OK, IFTTT can ~~access~~ Turn off my oven

5. Here is a nice overprivileged rule-specific ~~bearer token~~

We introduce XTokens (transfer tokens)

Mint a rule-specific token non-interactively

Does not increase the number of OAuth permission prompts

Measuring Channel-Online-Service Overprivilege in IFTTT

Channel connection issues	Opaque OAuth scopes	Many Private APIs
128/297 connected	107/128	69/128 online services have public APIs

Capture OAuth tokens of the same scope as that of IFTTT, and then exhaustively test online service APIs

OAuth
Online Service

Server-to-Server Communication

GET http://service1.com/?arg_a=1

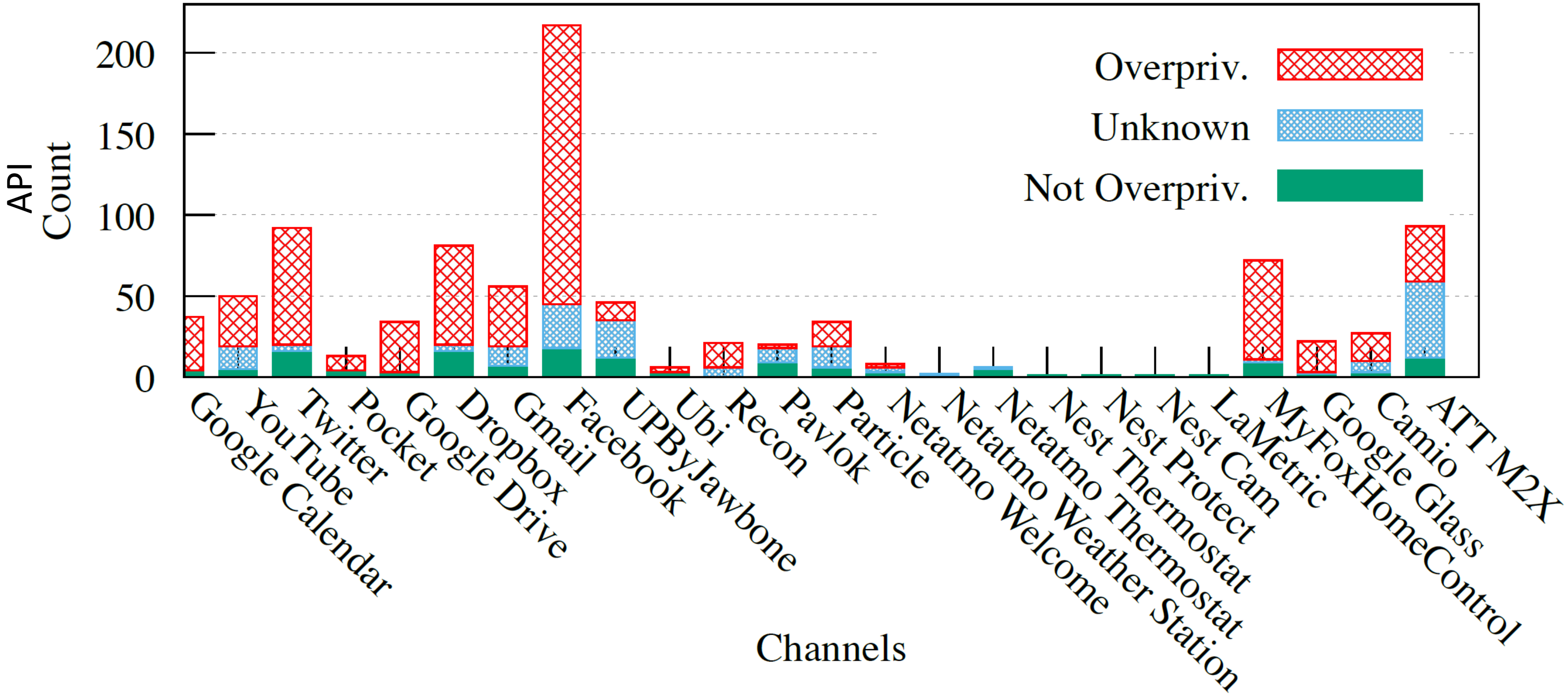
POST http://service2.com BODY
arg_a = 1

Inconsistent API Forms

String, Integer, Custom JSON, …

Input args are very diverse

75% of studied IFTTT Channels are Overprivileged



- 16 IoT and 8 Non-IoT channels studied; 18/24 overprivileged
- Covered 80.4% (46, 354/57, 632) of all recipes involved in 69 measurable channels

Lessons from IFTTT Analysis

- Channel Abstraction: good balance in usability-security tradeoff
    - But, leads to highly-privileged tokens inside IFTTT's infrastructure
- Highly-privileged tokens == Long-term security risk
    - Bearer tokens are known to be vulnerable to compromise
    - E.g., 4 channels vulnerable to open-redirector attack, 22 vulnerable to downgrade-only attack
- Overprivileged tokens == really bad idea