technische universität
dortmund

LS 12 /
TU Dortmund

FLASH: Ein on-premise Workflow Automation Service im Kontext Smart Home - Prototypische Implementierung basierend auf dem Eclipse Open IoT Stack

Konstantin Tkachuk

Abschluss-Vortrag

WS 16/17

Gliederung

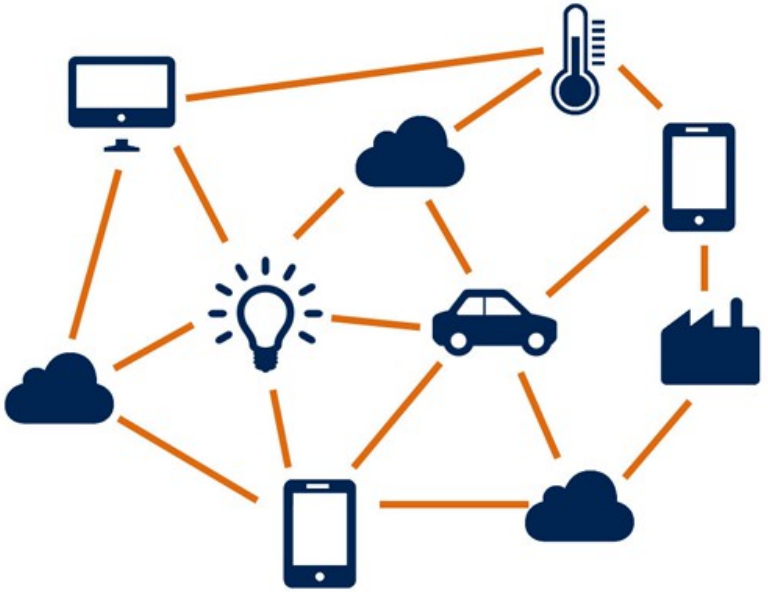
- Inhaltverzeichnis
- Motivation u. Grundlagen
  - Internet of Things
  - Smart Home
  - Task Automation Services
- Ziel(-erfüllung)
- Implementierung
- Evaluation
- Zusammenfassung & Ausblick

Hype cycle


expectations
Advanced Analytics With Self-Service Delivery
Autonomous Vehicles
Internet of Things
Speech-to-Speech Translation
Machine Learning
Wearables
Cryptocurrencies
Consumer 3D Printing
Natural-Language Question Answering
Smart Advisors
Micro Data Centers
Digital Dexterity
Software-Defined Security
Neurobusiness
Citizen Data Science
Biochips
IoT Platform
Connected Home
Affective Computing
Smart Robots
3D Bioprinting Systems for Organ Transplant
Volumetric Displays
Human Augmentation
Brain-Computer Interface
Quantum Computing
Hybrid Cloud Computing
Augmented Reality
Enterprise 3D Printing
Gesture Control
Virtual Reality
Autonomous Field Vehicles
Cryptocurrency Exchange
Bioacoustic Sensing
People-Literate Technology
Digital Security
Virtual Personal Assistants
Smart Dust
As of July 2015
Innovation Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
time
Plateau will be reached in:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau

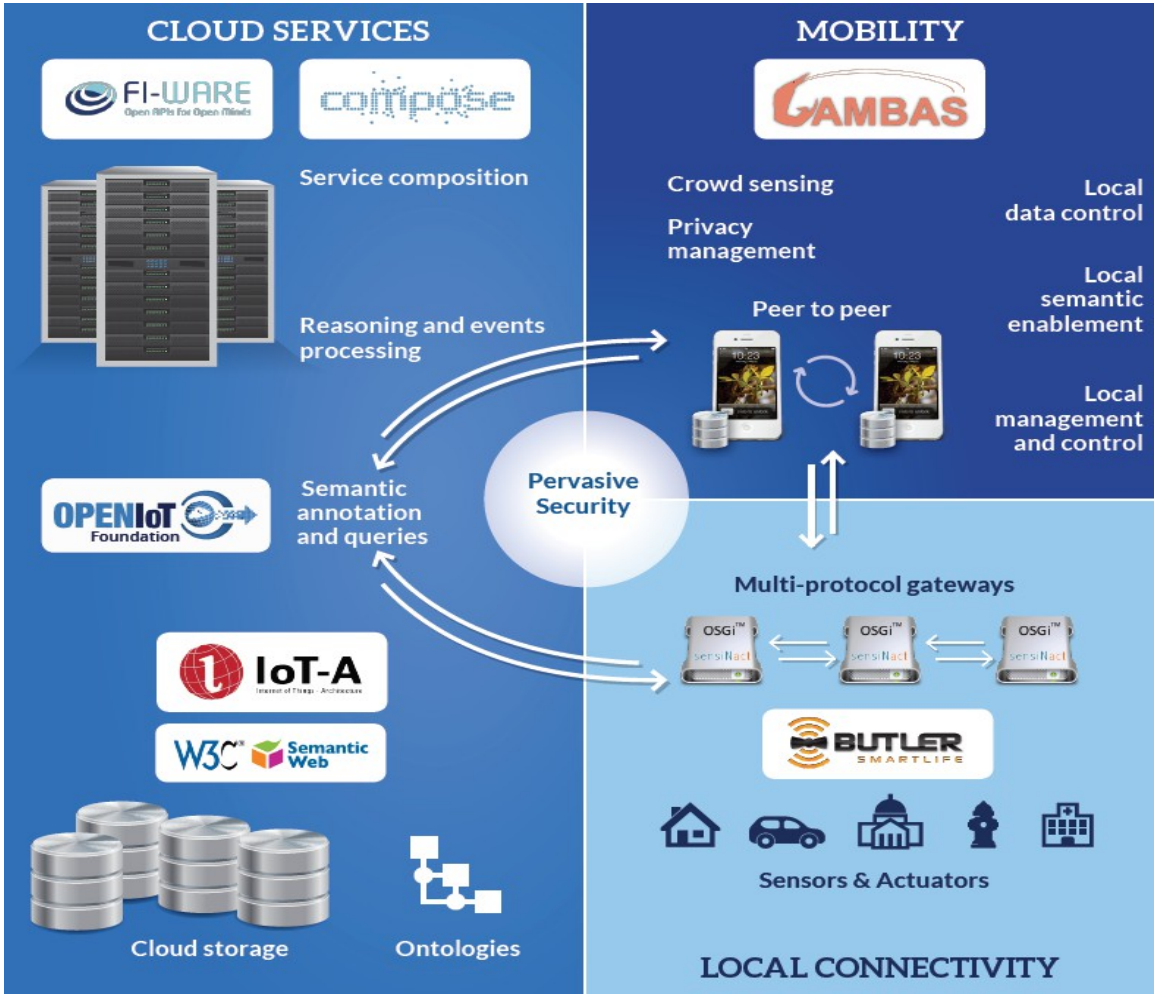
Internet der Dinge

- Begriff im Jahre 1999 von Kevin Ashton erstmals eingeführt
- Ermöglicht durch große Verbreitung von RFID, IEEE 802.11, Bluetooth, QR Codes, etc.
- Zahlreiche Forschungsprojekte vom IERC, ITU und weiteren
- Suche nach Mehrwert durch Kombination mit unterschiedlichen anderen Technologien



Internet der Dinge

- IoT-A – Architektur als Grundlage für andere Projekte
- OpenIoT – Integration von IoT mit Cloud Computing durch Semantiken
- SmartAction – Projektübergreifende Zusammenarbeit

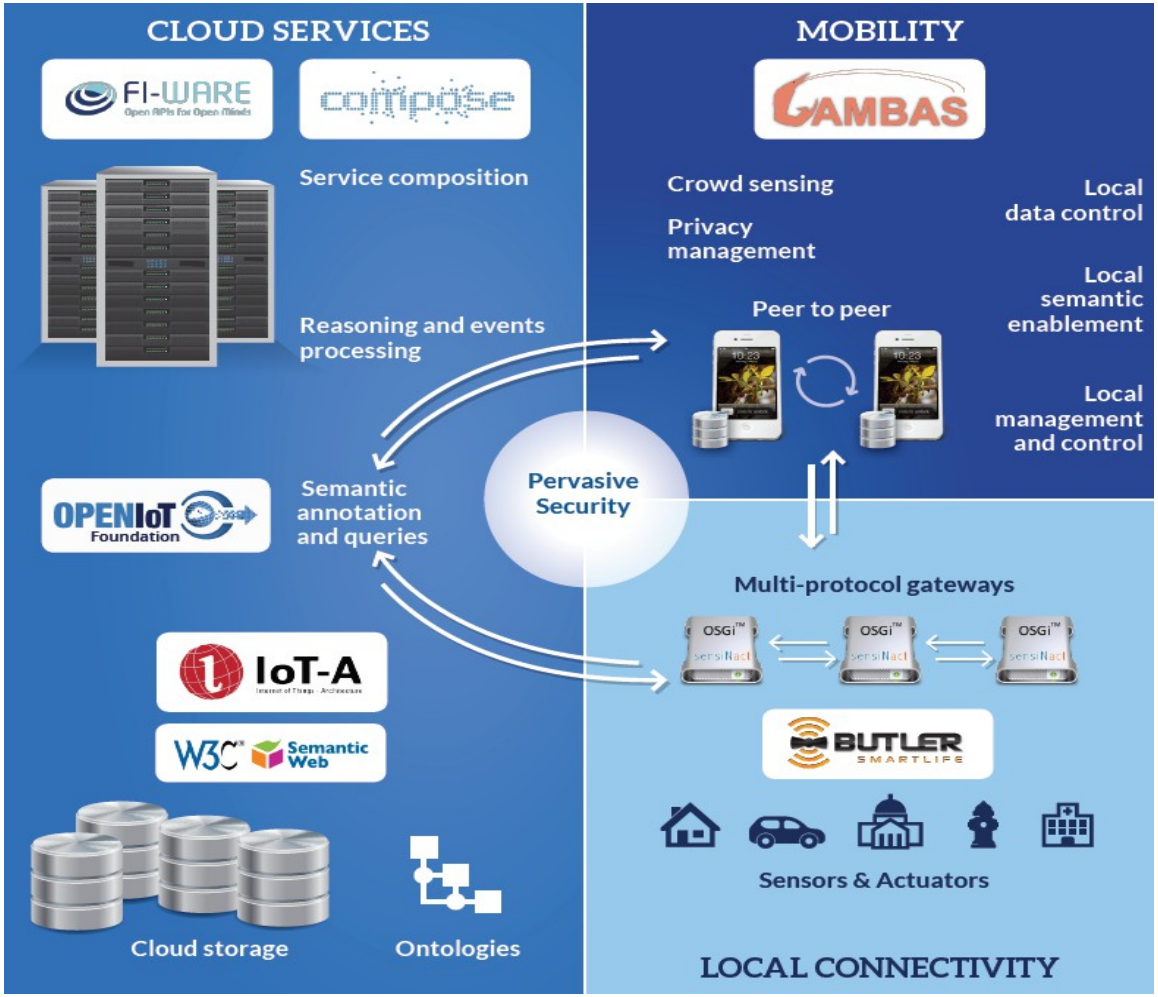


Internet der Dinge

- IoT-A – Architektur als Grundlage für andere Projekte
- OpenIoT – Integration von IoT mit Cloud Computing durch Semantiken
- SmartAction – Projektübergreifende Zusammenarbeit

Diese Arbeit:

Smart Home ↔ TAS

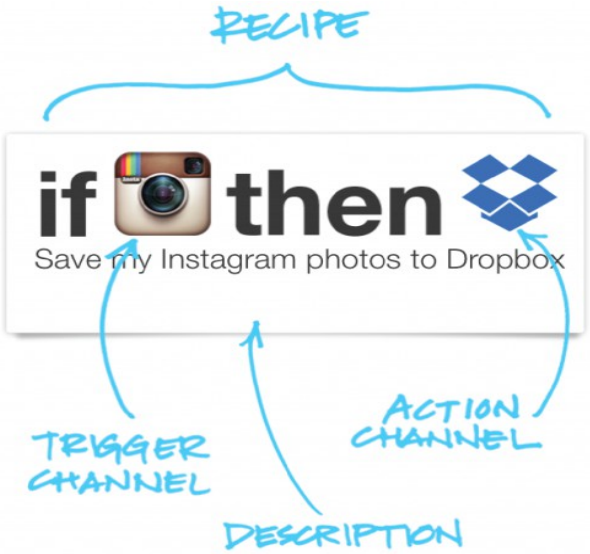


Smart Home

- Teil von IoT
- Konzentriert sich auf die direkte Steuerung von Geräten/Services im Kontext eines Hauses
- Kommerziell
  - Unterstützung von begrenzter Anzahl von Geräten
- Open Source
  - Fokus auf maximale Offenheit
  - Bedarf fortgeschrittener technischer Kenntnisse

Task Automation Services (TAS)

- Automatisierung historisch eines der zentralen Ziele der Menschheit – von Mühlen bis Kaffeeautomaten
- Heute: Automatisierung von Serviceinteraktionen
- Task Automation Service:
  - Dienst, der die Orchestration von anderen Diensten ermöglicht
- Typische Merkmale
  - Device/Web Channels
  - Event-Condition-Action Regeln
  - Visuelle Regel-Editoren
  - Device-/Webdriven Rule Execution Profiles



Task Automation Services – Überblick

		Web						Smartphone				Home	
		Ifttt	Zapier	CloudWork	Elastic.io	ItDuzzit	Wappwolf	On{x}	Tasker	Atooma	AutomateIt	WigWag	Webee
Channels	Web channel support	✓	✓	✓	✓	✓	Few	Few	✓	✓	✓	✓	✓
	Device channel support	Few	×	×	×	×	×	✓	✓	✓	✓	✓	✓
	Smartphone resources as channels	✓	×	×	×	×	×	✓	✓	✓	✓	×	×
	Public channels support	✓	×	×	✓	✓	×	✓	×	×	×	×	×
	Pipe channel support	×	×	×	✓	×	Few	Few	×	×	×	×	×
	Group channel support	×	×	×	×	×	×	×	×	×	×	Few	×
	Device channel discovery	×	×	×	×	×	×	×	×	×	×	✓	Few
Rules	Multi-event rules	×	×	×	×	×	×	✓	✓	×	×	✓	×
	Multi-action rules	×	×	×	✓	×	×	✓	✓	×	✓	✓	✓
	Chain rules	×	×	×	✓	×	Few	Few	×	×	×	×	×
	Group rules	×	×	×	×	×	×	×	×	×	×	Few	×
	Collision handling	×	×	×	×	×	×	×	×	×	×	×	×
	Predefined common rules	×	×	✓	×	✓	✓	×	×	×	✓	✓	✓
	Rule execution profile	WD	WD	WD	WD	WD	WD	DD	DD	DD	DD	DD	DD
TAS	Visual rule editor	✓	✓	×	✓	✓	✓	×	✓	✓	✓	✓	✓
	Provides API	×	✓	×	✓	×	×	×	×	×	×	×	×
	Programming language	×	×	×	✓	×	×	✓	Few	×	✓	✓	✓

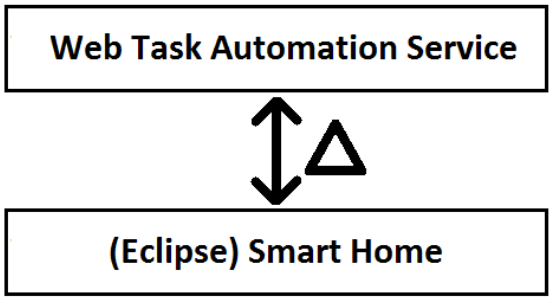
* ✓ = supported; × = not supported; Few = few support; WD = Web-driven execution profile; and DD = device-driven execution profile.

3 unterschiedliche Typen

1. Web TAS
   - Automatisierung von Web-Serviceinteraktionen in der Cloud
2. TAS auf Smartphone
   - Automatisierung von App-Interaktionen
3. (Smart-)Home
   - Automatisierung von Geräten

Vergleich SmartHome und TAS

- Gemeinsamkeiten von Webbasierten TAS und SmartHome
  - Automatisierung von Geräten und Services
  - Event-Condition-Action Regeln kommen zum Einsatz
  - Visuelle Regeleditoren
- Unterschiede
  - Fokus der Automatisierung
  - On-premise vs. Cloud



Bis dato waren webbasierte TAS und SmartHome völlig unabhängig. Nun wurden sie kombiniert.

Mehrwert durch Kombination

- Zugriffsdaten werden lokal gelagert
  → Datenschutz

- Geräte werden lokal gesteuert
  → Funktionalität im Haus bleibt bei Verbindungsabbruch erhalten
  → Reaktionszeiten von Geräten verbessert

- Komplexere Regeln/Szenarien ermöglichen
  → Wiederverwendung von existierenden Funktionalitäten

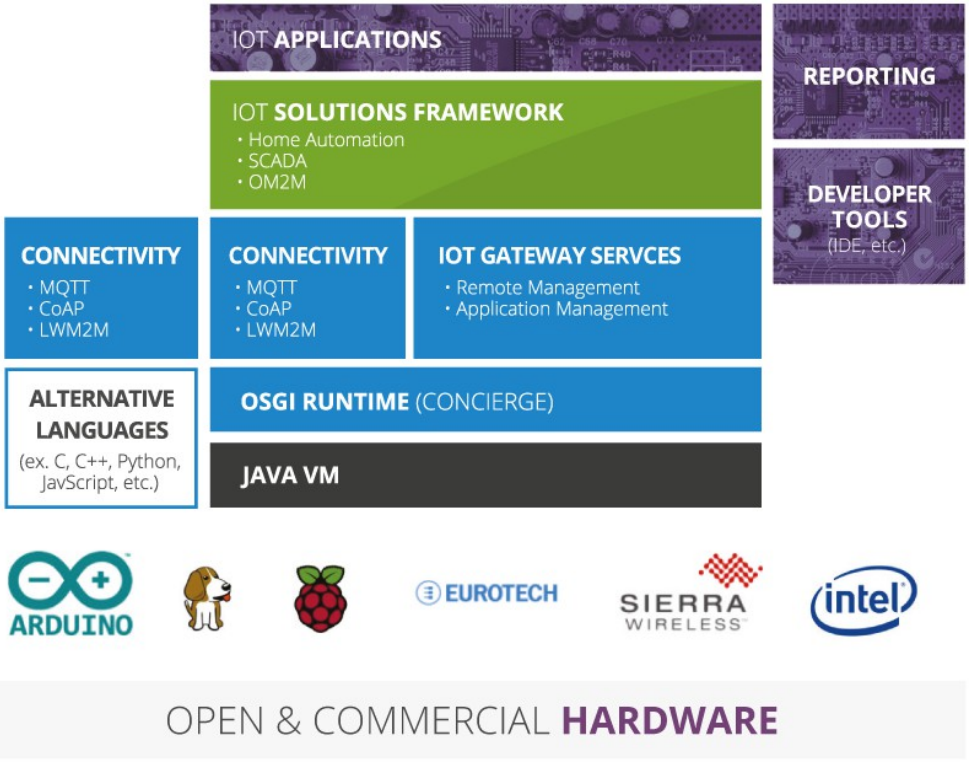
Ziele der Arbeit

- Demonstrator entwickeln, der
  - Geräte im Haus direkt steuern kann ✓
  - Bei vorhandener Internetverbindung ausgewählte Webservices automatisieren kann ✓
  - Sämtliche Zugriffsdaten lokal speichert ✓
  - Die Webservices in die Rule Engine integriert
    → komplexe Szenarien ✓
  - Funktionalität in Demo Regeln demonstriert ✓
  - GUI: Eine Möglichkeit bietet neue Regeln zur Laufzeit hinzuzufügen ✓
  - On-premise auf einem Raspberry Pi 3 Model B läuft ✓

- Evaluation des Demonstrators

Implementierung - Eclipse SmartHome

- Framework zur Entwicklung von eigenen SmartHome Lösungen
- Unterstützt Entwickler bei:
  - Discovery-Services
  - Bindings
  - Rule Engine
  - GUI
  - …
- Wird in bekannten Lösungen eingesetzt, z.B. openHAB, Qivicon

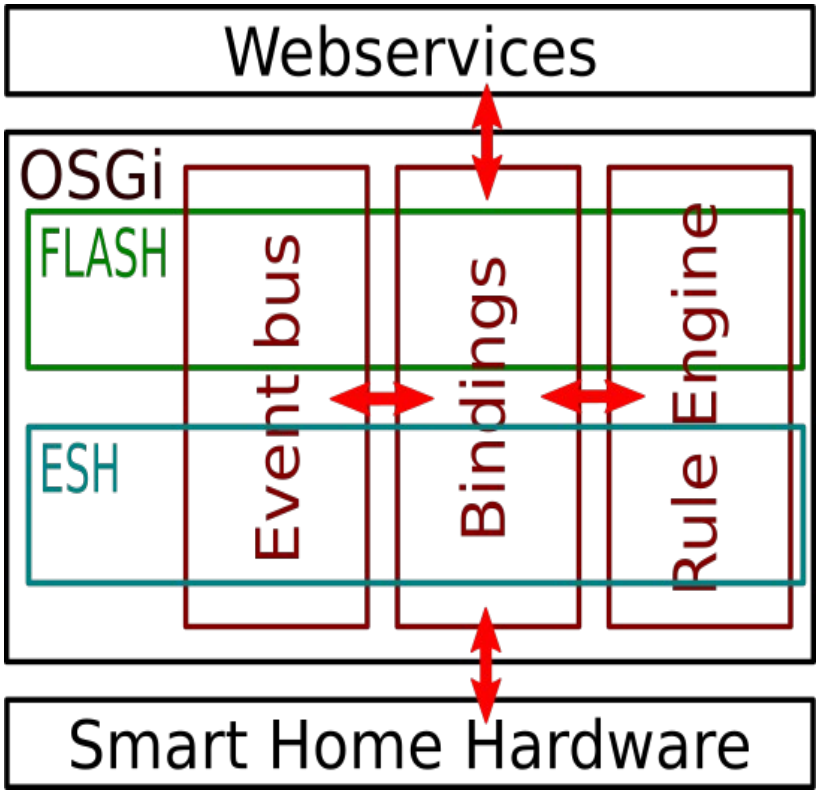


Implementierung

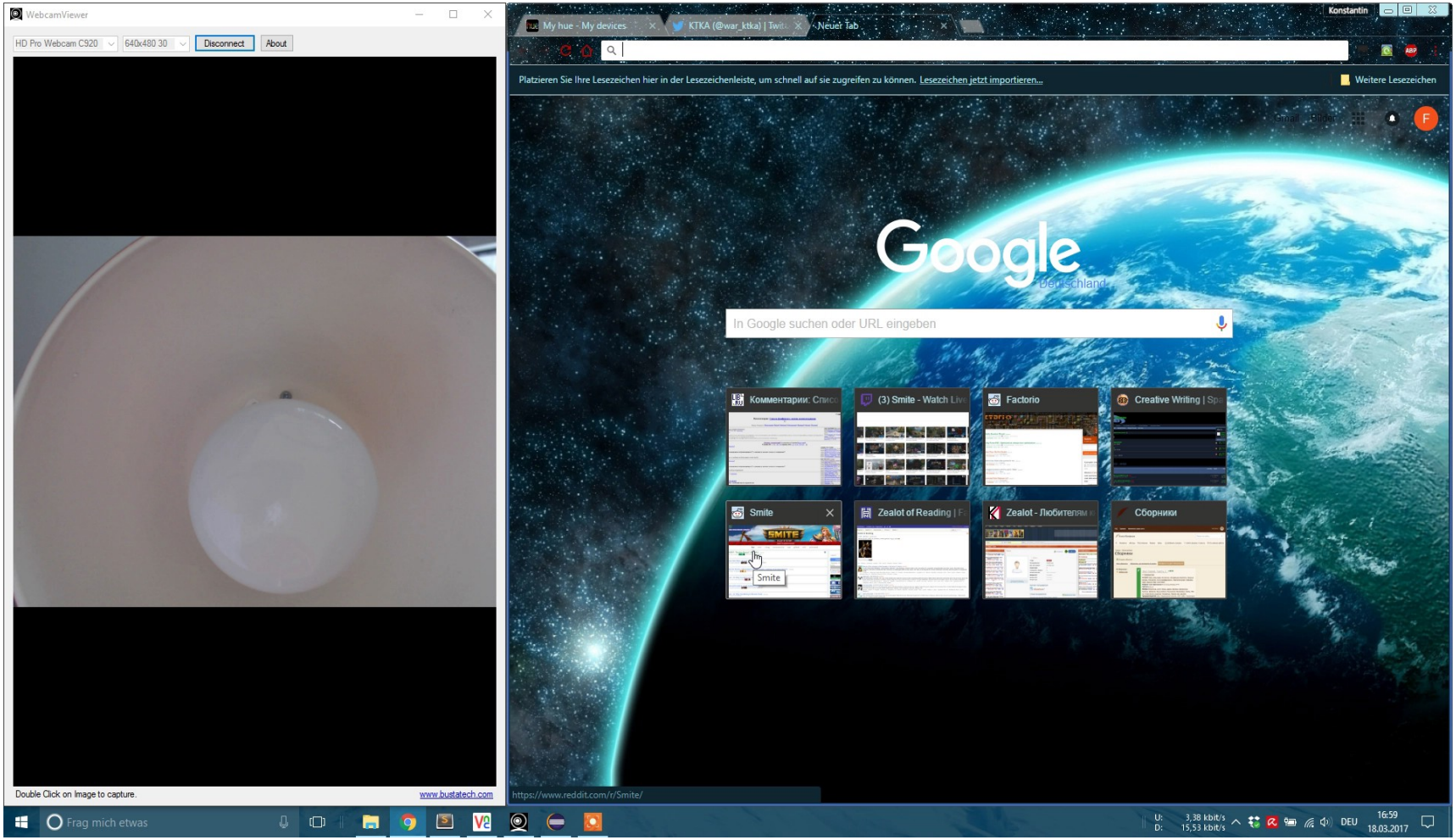
- ESH besteht aus >130 Bundles
- FLASH derzeit aus 7 weiteren

- Integration von Webservices durch entsprechende Bindings

  → Einholen neuer Informationen über Polling oder WebHooks
- Integration in Rule Engine durch Wiederverwendung existierender und Definition neuer Trigger/Conditions/Actions
- Definition neuer Regeln über JSON



Implementierung Video



Evaluation – Vergleich mit IFTTT und Zapier

- Allgemein
  - Mächtigkeit der Rule Engines
  - Datenschutz
- Datendurchsatz
  - Verschieben von Dateien in der Dropbox
  - Messung der Dauer
- Reaktionszeiten
  - Steuerung einer Philips Hue Lampe bei Tweet

Evaluation – Allgemein

		FLASH	Web	
			Ifttt	Zapier
Channels	Web channel support	✓	✓	✓
	Device channel support	✓	Few	×
	Smartphone resources as channels	×	✓	×
	Public channels support	✓	✓	×
	Pipe channel support	✓	×	×
	Group channel support	×	×	×
	Device channel discovery	✓	×	×
Rules	Multi-event rules	✓	×	×
	Multi-action rules	✓	×	×
	Chain rules	×	×	×
	Group rules	×	×	×
	Collision handling	×	×	×
	Predefined common rules	✓	×	×
	Rule execution profile	DD	WD	WD
TAS	Visual rule editor	×	✓	✓
	Provides API	✓	×	✓
	Programming language	✓	×	×

* ✓ = supported; × = not supported;

FLASH:

- Rule Engine mächtiger, als IFTTT und Zapier

  → deutlich Komplexere Regeln möglich

- Visueller Regeleditor fehlt – neue Regeln über JSON

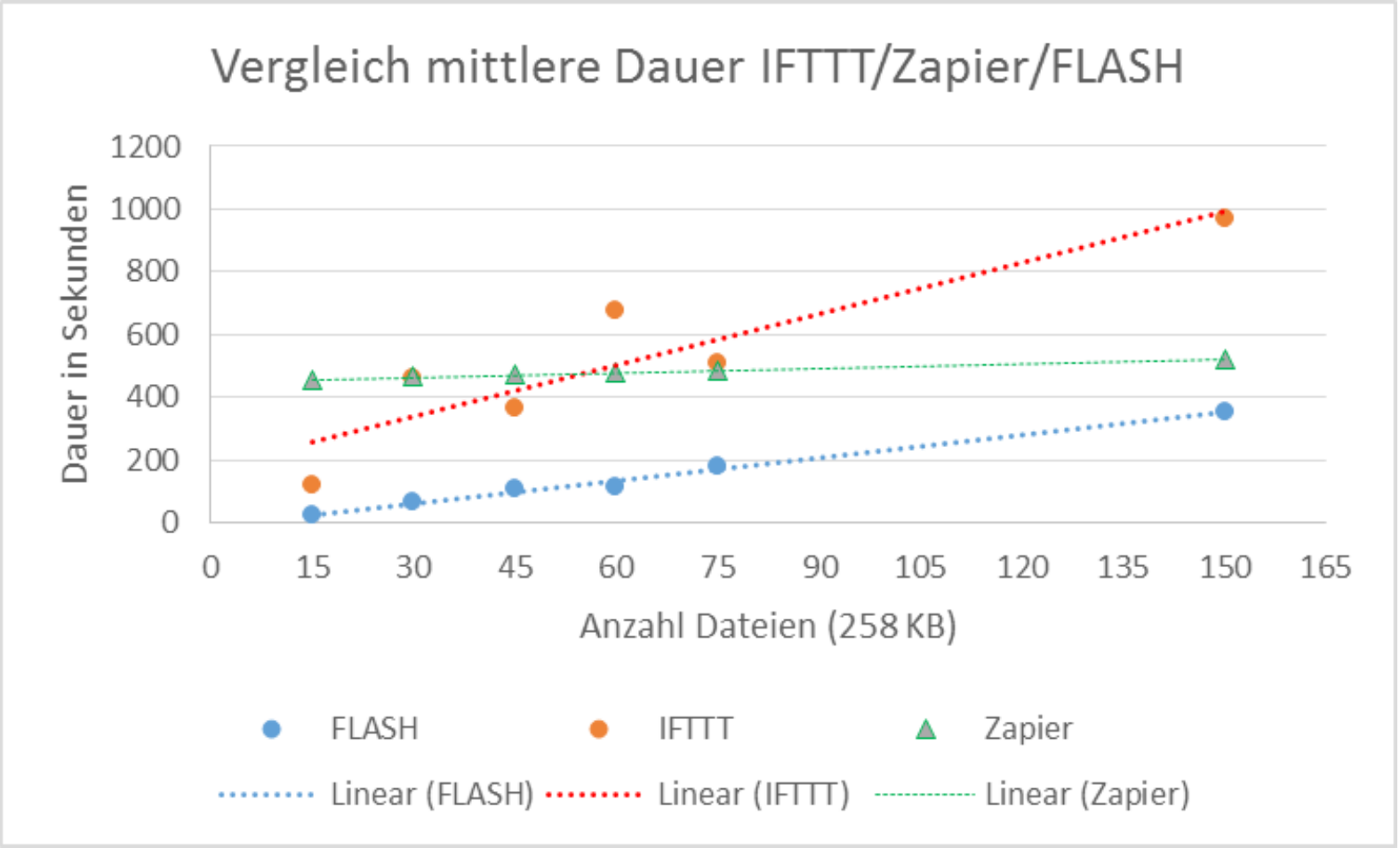
- DD vs. WD Ausführungsprofil

  → keine arbiträren Eingrenzungen

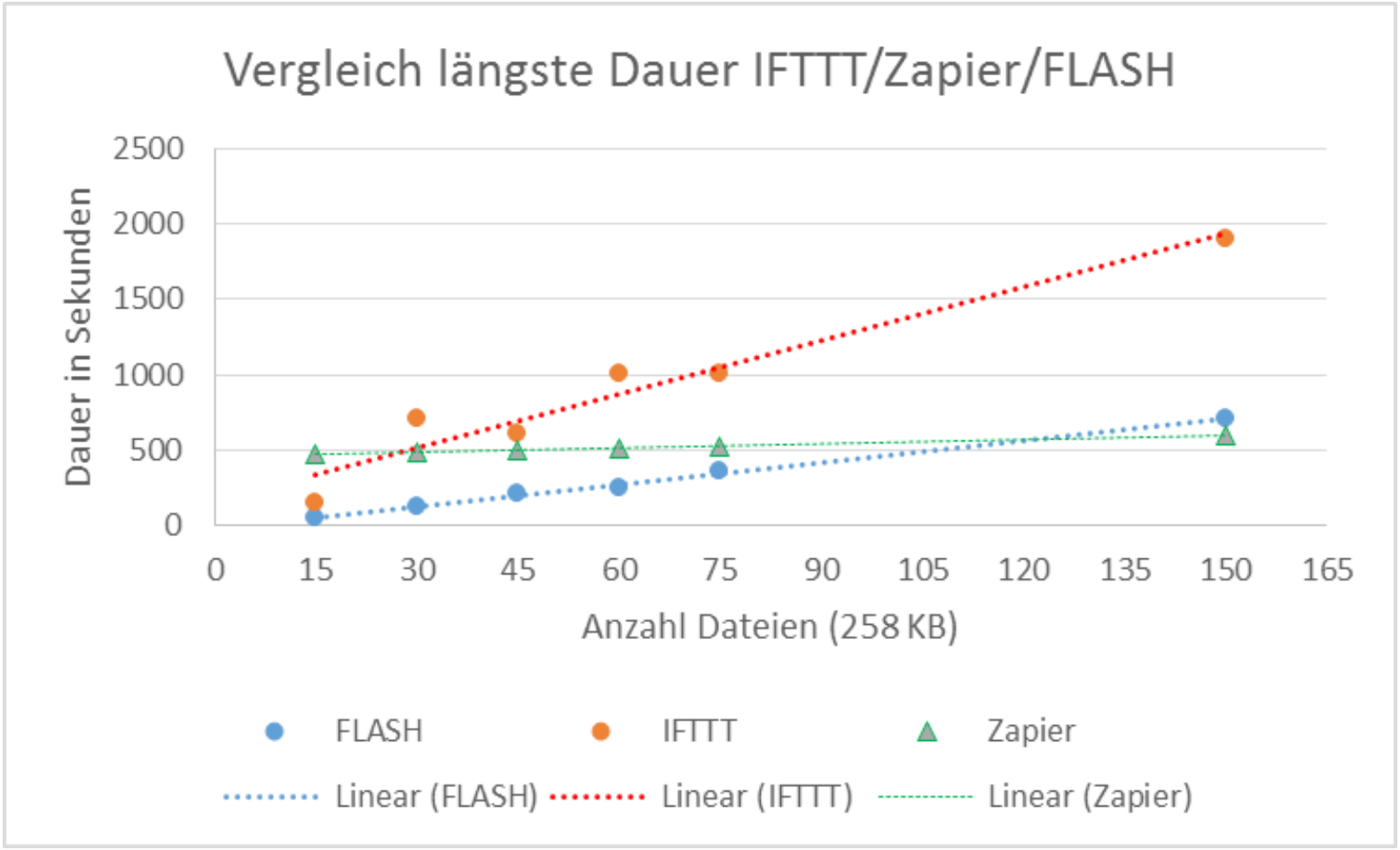
- Sämtliche Zugriffsdaten werden lokal gelagert

  → Datenschutz

Evaluation – Verschieben von Dateien in Dropbox



Evaluation – Verschieben von Dateien in Dropbox



Evaluation – Steuern von physchen Geräten

- Reaktionszeit bei Steuern einer Philips Hue Lampe bei Tweet

  FLASH: ≈ 1 Sekunde

  IFTTT: >10 Sekunden

  Zapier: -

Zusammenfassung & Ausblick

- Durch Kombination von Smart Home und Task Automation Services lässt sich Mehrwert erzielen:
  - Bessere Reaktionszeiten
  - Zugriffsdaten nur lokal
  - Vergleichbarer Datendurchsatz
- Eclipse SmartHome ist gut als Grundlage für weitere Lösungen geeignet

- Mögliche Erweiterungen:

  Visueller Regeleditor, Kollisionsbehandlung, gemeinsames Backend, bessere Datensicherung (Backups), ...