Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Privacy Assessment using Static Taint Analysis (Tool Paper)

Marcel von Maltitz, Cornelius Diekmann, Georg Carle

June 20th 2017

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

What is Privacy?

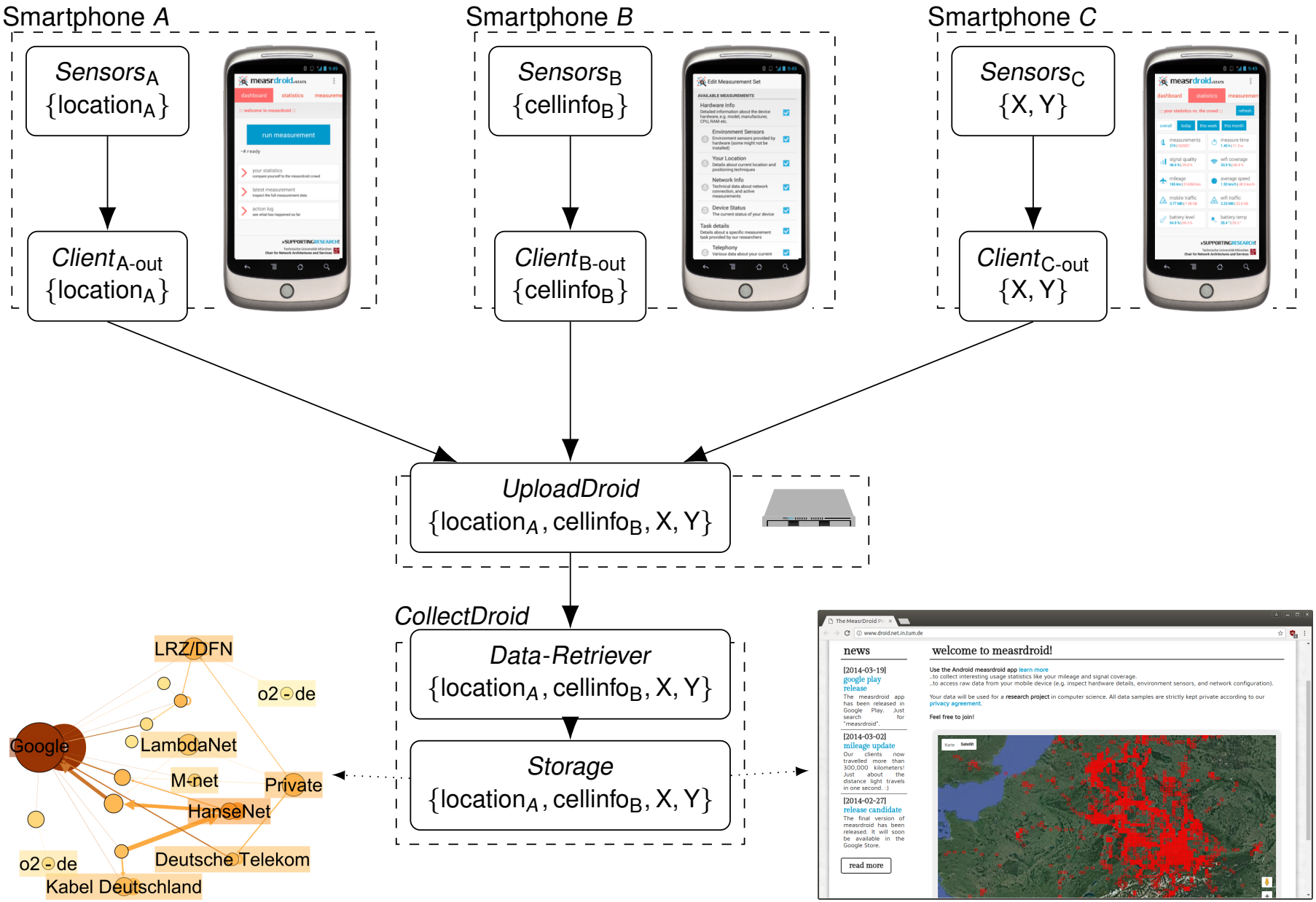
- Unlinkability
  - Impossibility to combine present information gain further information
- Transparency
  - Users can gain insight into processes and software architectures
- Intervenability
  - Data subjects (users): be in control over their data and exercise their user rights
  - Process owners (providers): be in control of their technical systems

What is Privacy?

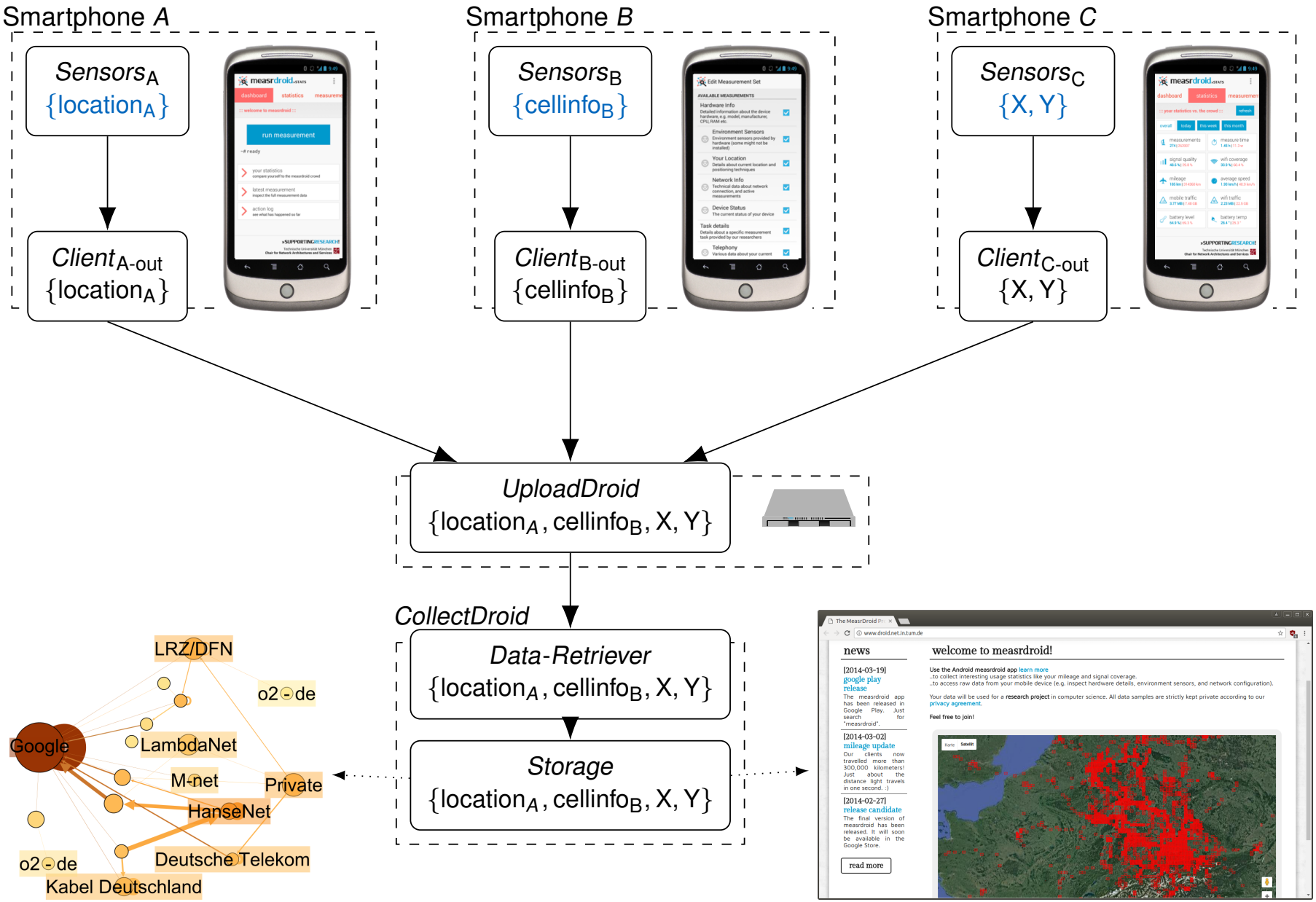
- Unlinkability
  - Impossibility to combine present information gain further information
- Transparency
  - Users can gain insight into processes and software architectures
- Intervenability
  - Data subjects (users): be in control over their data and exercise their user rights
  - Process owners (providers): be in control of their technical systems

Compatible with Pfitzmann and Rost [11], Steinbrecher [12], Bock and Rost [2], Bock, Rost, and Pfitzmann [5], German Standardized Data Protection Model [1], Global Privacy Standard [3], Privacy by Design [4], Hoepman [10]

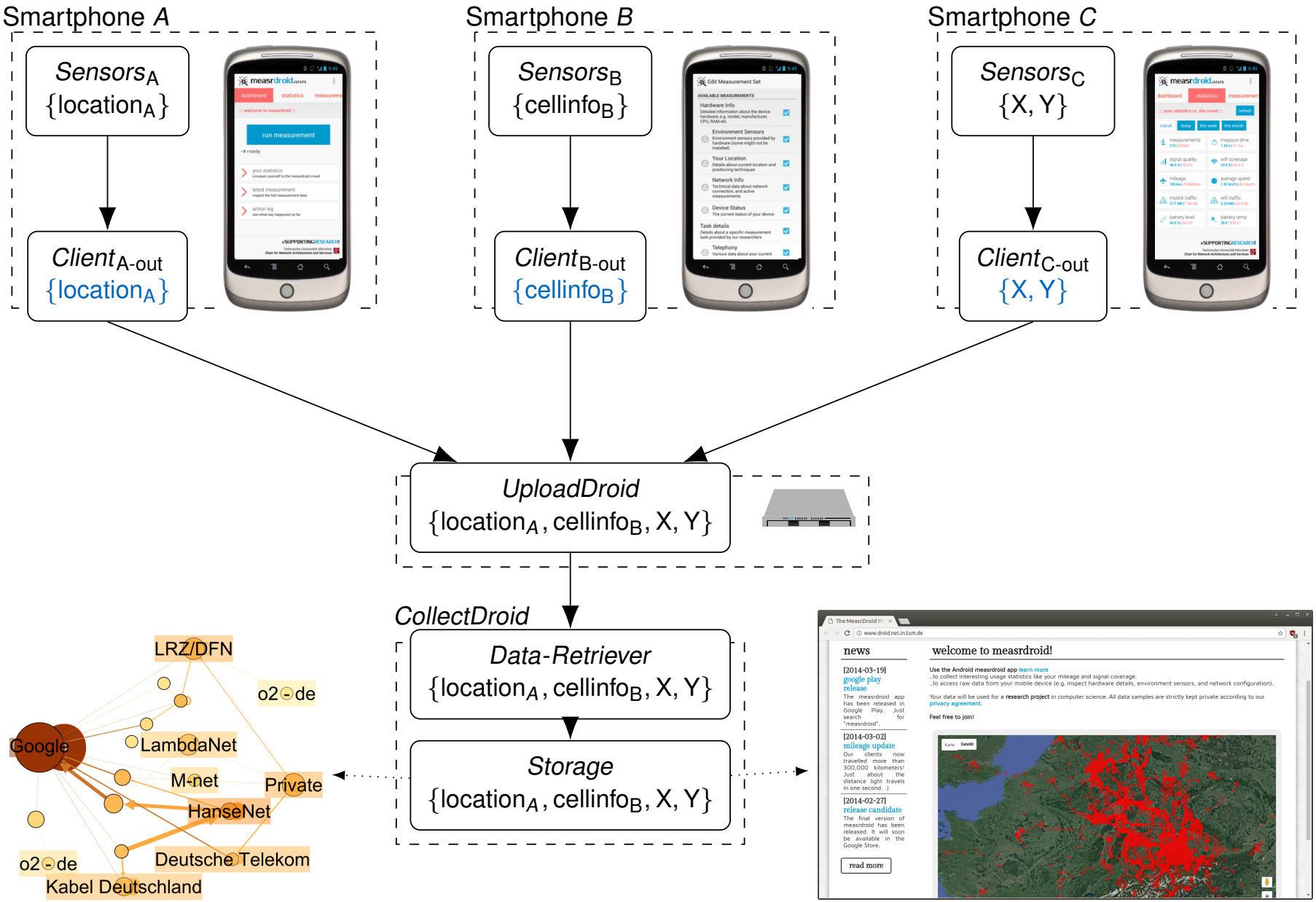
Example Scenario



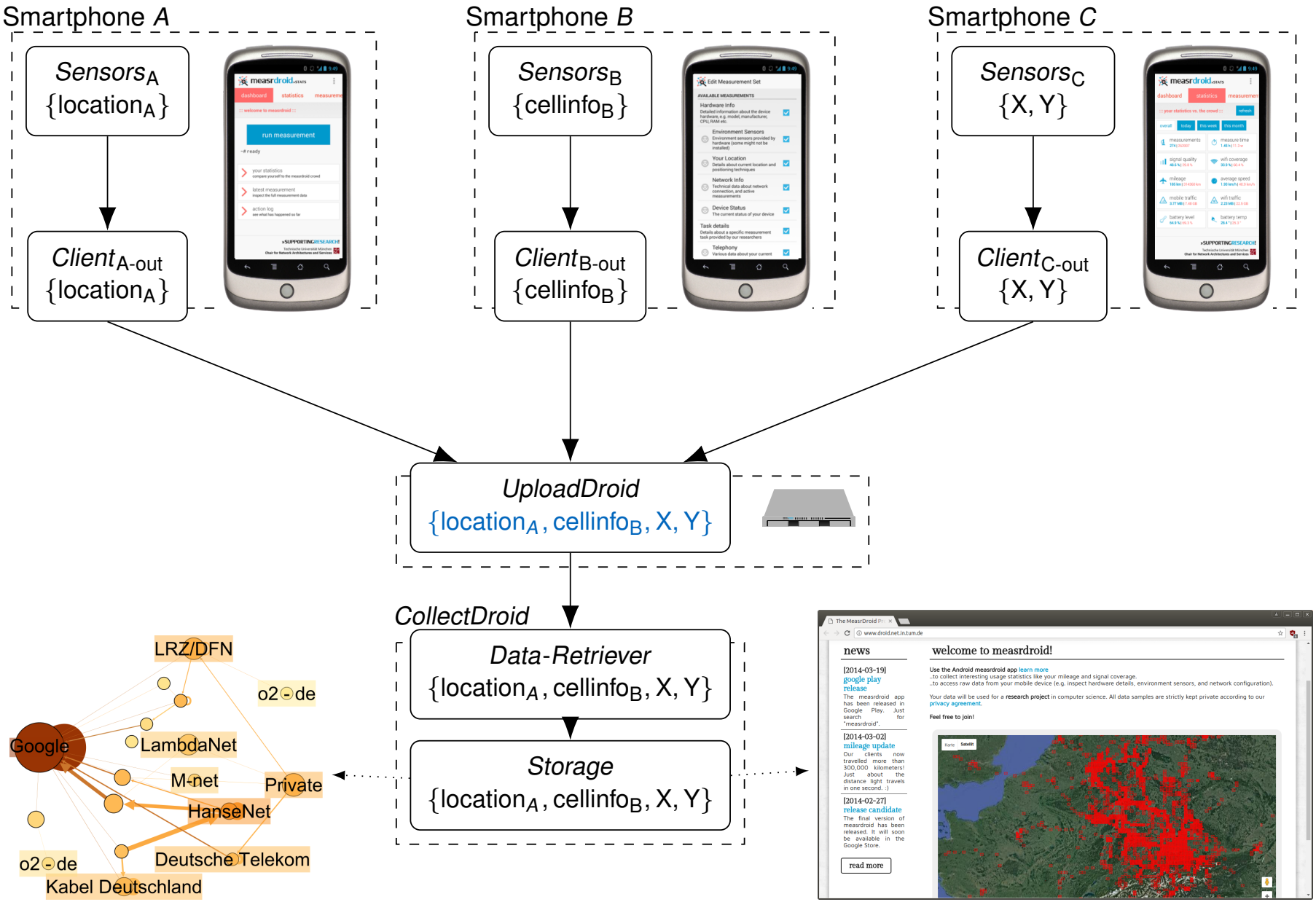
Example Scenario



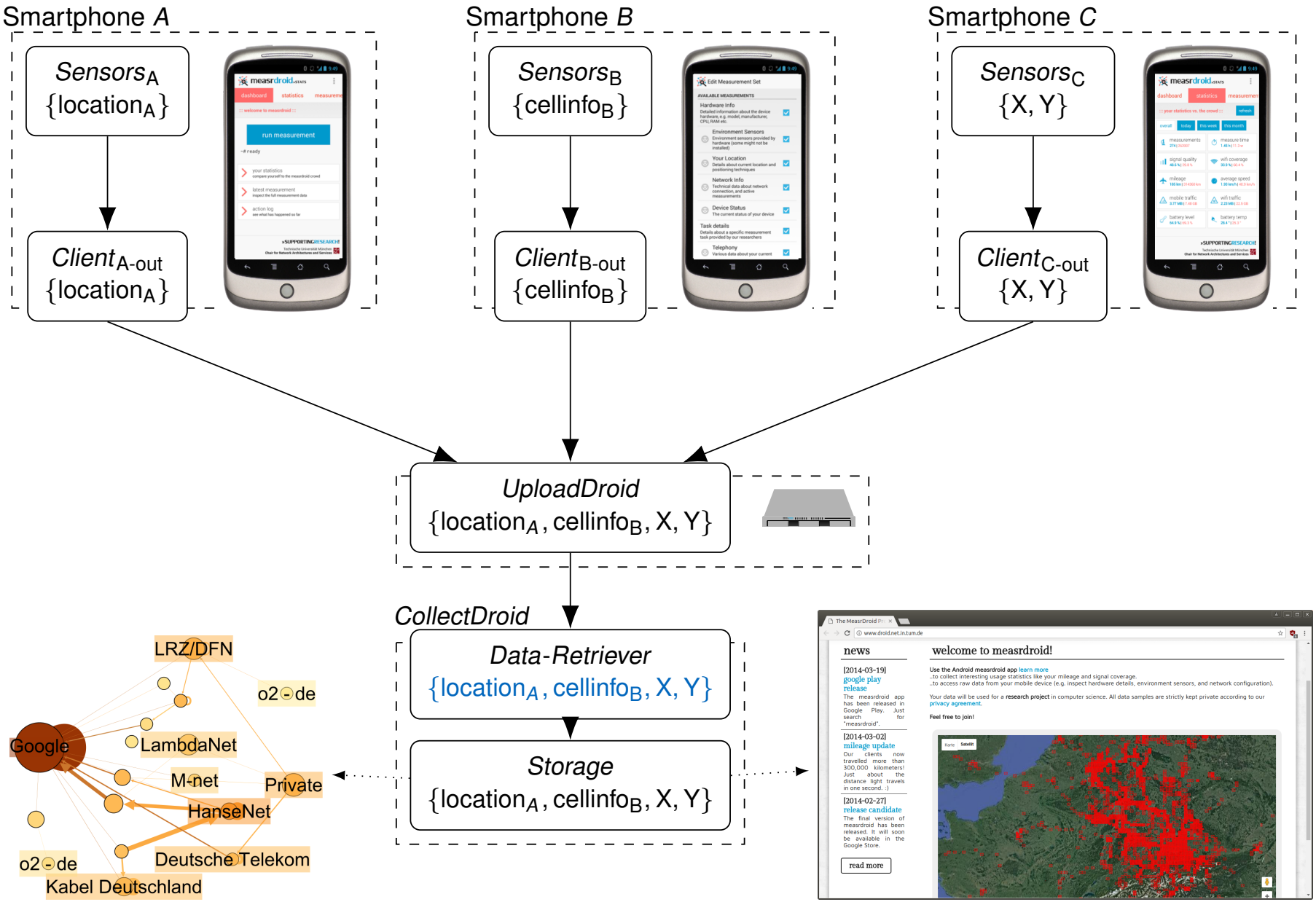
Example Scenario



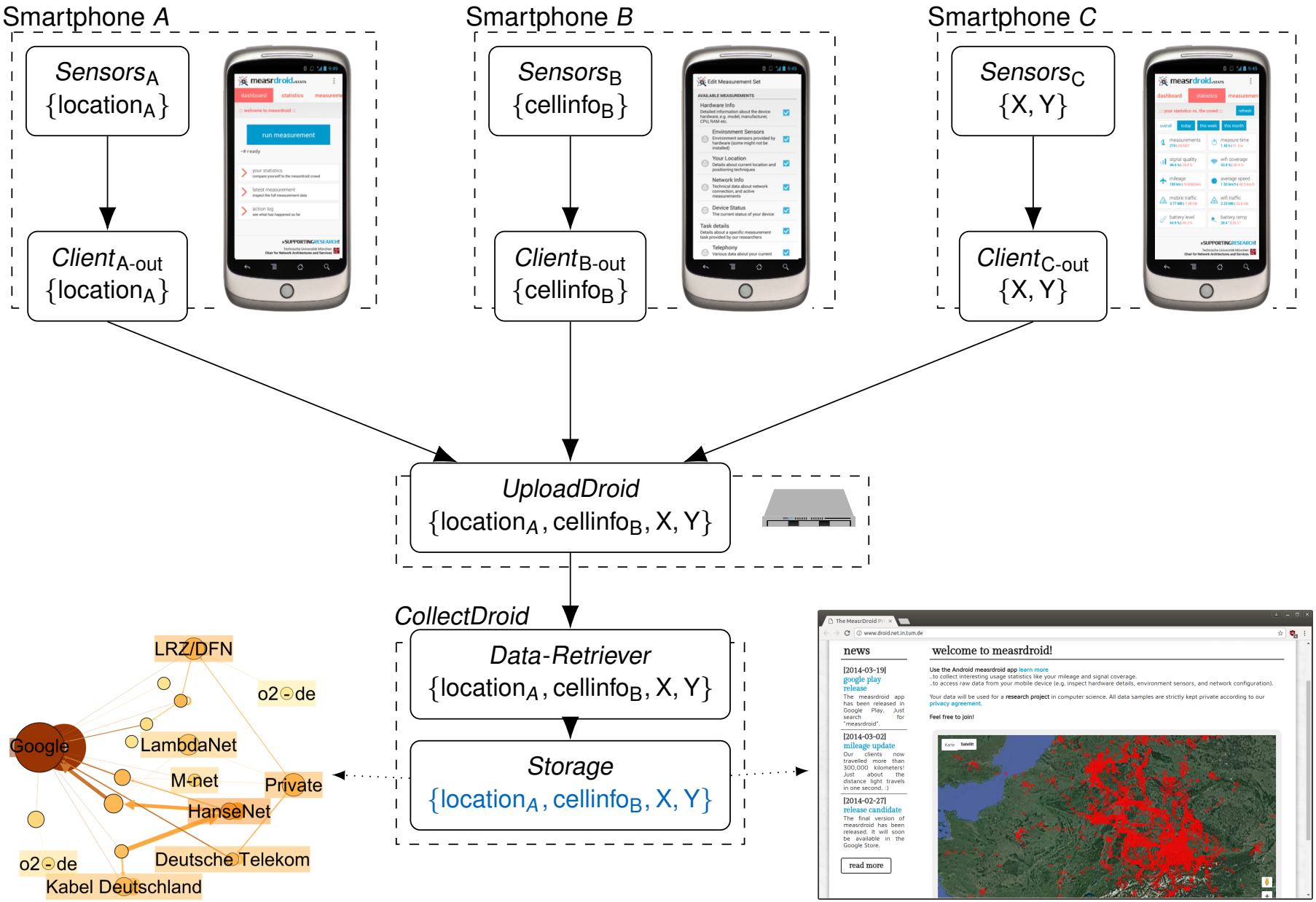
Example Scenario



Example Scenario



Example Scenario



Formalizing Taint Analysis

- Graph $G = (V, E)$
- Total function $t :: \mathcal{V} \Rightarrow \textit{taintlabel set}$
  - Example: $t\ \textit{Storage} = \{\text{location}_A, \text{cellinfo}_B, \text{X}, \text{Y}\}$

$$\text{tainting}\ (V, E)\ t \equiv \forall v \in V.\ \forall r \in \{r.\ (v, r) \in E^+\}.\ \ t\ v \subseteq t\ r$$

Formalizing Taint Analysis

- Graph $G = (V, E)$
- Total function $t :: \mathcal{V} \Rightarrow \textit{taintlabel set}$
  - Example: $t\ \textit{Storage} = \{\text{location}_A, \text{cellinfo}_B, \text{X}, \text{Y}\}$

$$\text{tainting}\ (V, E)\ t \equiv \forall v \in V.\ \forall r \in \{r.\ (v, r) \in E^+\}.\ \ t\ v \subseteq t\ r$$

- project $a\ Ts \equiv$ **if** $a \in Ts$ **then** confidential **else** unclassified

Formalizing Taint Analysis

- Graph $G = (V, E)$
- Total function $t :: \mathcal{V} \Rightarrow \textit{taintlabel set}$
  - Example: $t\ \textit{Storage} = \{\text{location}_A, \text{cellinfo}_B, \text{X}, \text{Y}\}$

$$\text{tainting}\ (V, E)\ t \equiv \forall v \in V.\ \forall r \in \{r.\ (v, r) \in E^+\}.\ \ t\ v \subseteq t\ r$$

- project $a\ Ts \equiv$ **if** $a \in Ts$ **then** confidential **else** unclassified

$$\text{tainting}\ G\ t \longleftrightarrow \forall a.\ \text{bell-lapadula}\ G\ (\text{project}\ a\ \circ\ t)$$

Formalizing Taint Analysis

- Graph $G = (V, E)$
- Total function $t :: \mathcal{V} \Rightarrow \textit{taintlabel set}$
  - Example: $t\ \textit{Storage} = \{\text{location}_A, \text{cellinfo}_B, \text{X}, \text{Y}\}$

$$\text{tainting}\ (V, E)\ t \equiv \forall v \in V.\ \forall r \in \{r.\ (v, r) \in E^+\}.\ \ t\ v \subseteq t\ r$$

- project $a\ Ts \equiv$ **if** $a \in Ts$ **then** confidential **else** unclassified

$$\text{tainting}\ G\ t \longleftrightarrow \forall a.\ \text{bell-lapadula}\ G\ (\text{project}\ a\ \circ\ t)$$

- Also works with trusted entities and untainting!

Formalizing Taint Analysis

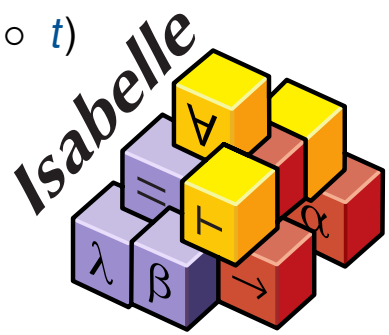
- Graph $G = (V, E)$
- Total function $t :: \mathcal{V} \Rightarrow \textit{taintlabel set}$
  - Example: $t\ \textit{Storage} = \{\text{location}_A, \text{cellinfo}_B, \text{X}, \text{Y}\}$

$$\text{tainting}\ (V, E)\ t \equiv \forall v \in V.\ \forall r \in \{r.\ (v, r) \in E^+\}.\ \ t\ v \subseteq t\ r$$

- project $a\ Ts \equiv$ **if** $a \in Ts$ **then** confidential **else** unclassified

$$\text{tainting}\ G\ t \longleftrightarrow \forall a.\ \text{bell-lapadula}\ G\ (\text{project}\ a \circ t)$$

- Also works with trusted entities and untainting!

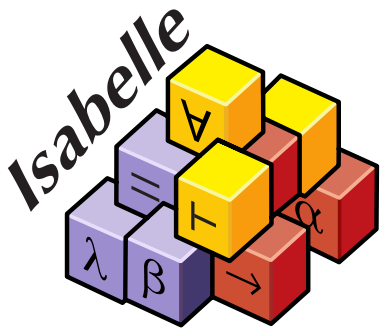


Approach

- Get your information flow as graph
- Add taint labels
- Do privacy analysis

Approach

- Get your information flow as graph
- Add taint labels
- Do privacy analysis



Rest of this talk: Auditing the real System!

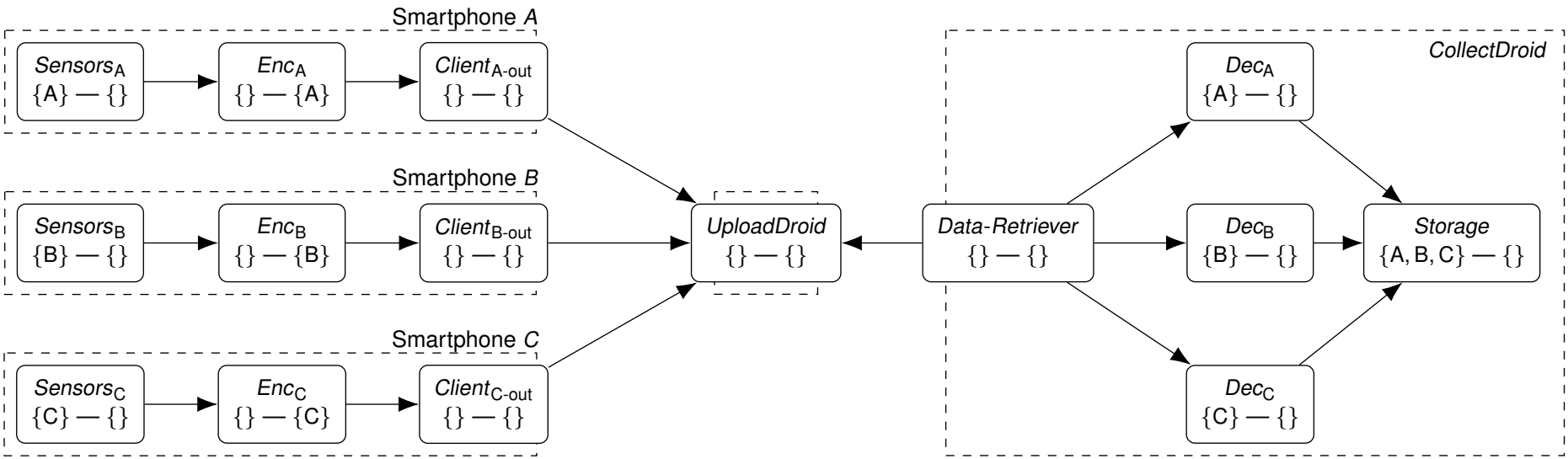
- Fully verified with Isabelle/HOL
- From abstract privacy concept to low level implementation
- Probably largest fully-verified real-world privacy audit of a firewall
- One example of a whole class of applications

Tools:

- topoS: policy analysis framework, enhanced with tainting
- fffuu: firewall analysis

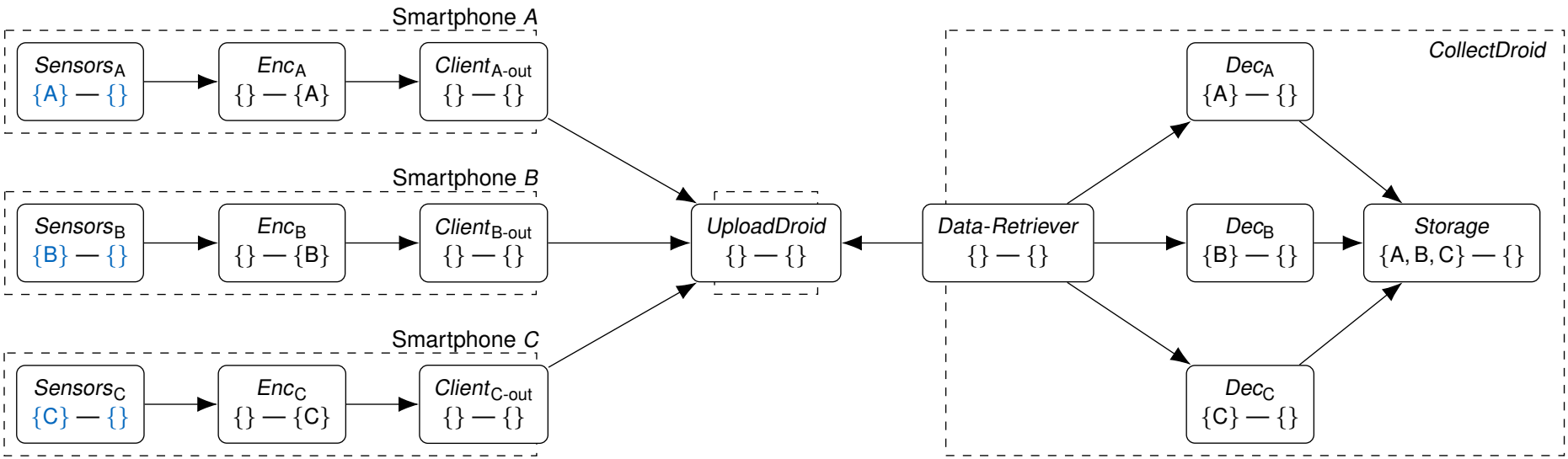
MeasrDroid – Architecture Formalized

- System for collecting smartphone sensor data
- Deployed at TUM
- Syntax: {taints}—{untaints}



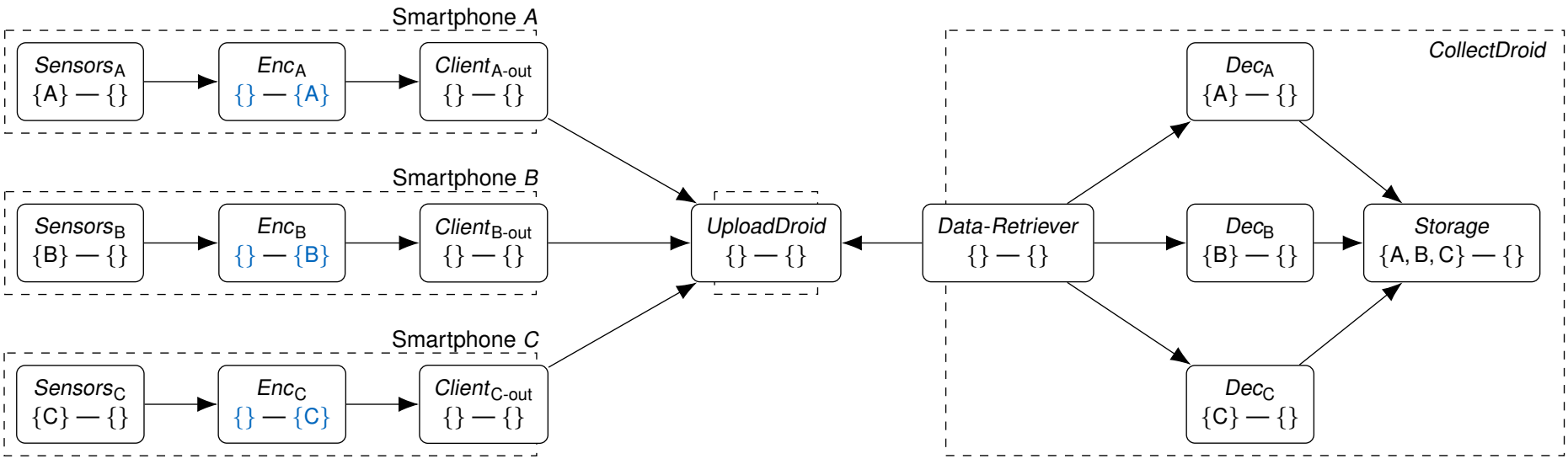
MeasrDroid – Architecture Formalized

- System for collecting smartphone sensor data
- Deployed at TUM
- Syntax: {taints}—{untaints}



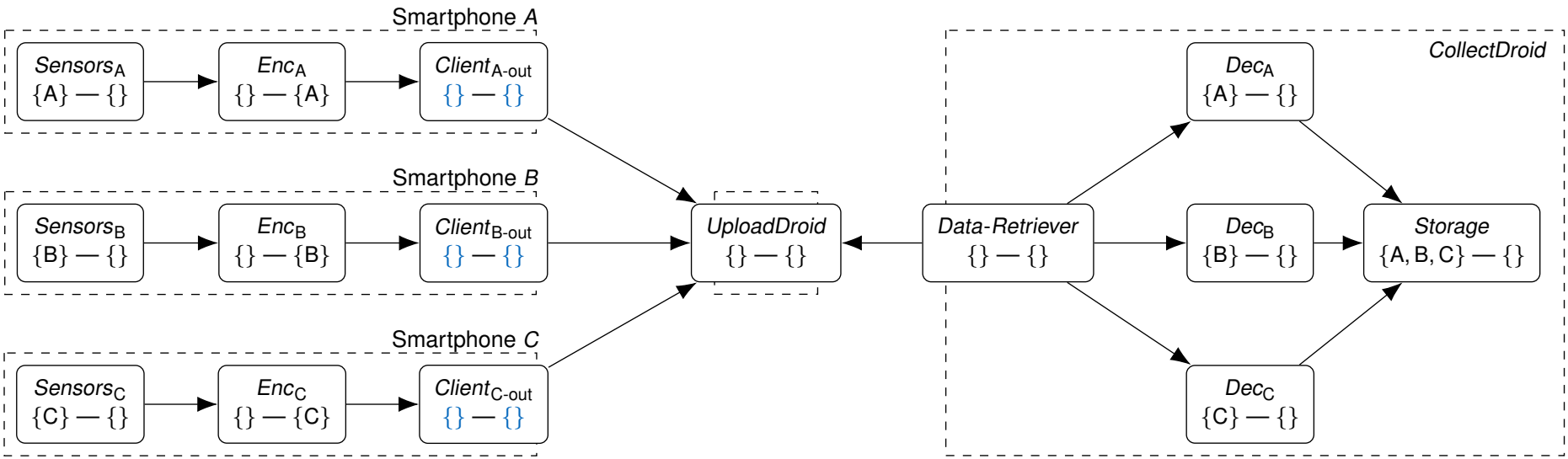
MeasrDroid – Architecture Formalized

- System for collecting smartphone sensor data
- Deployed at TUM
- Syntax: {taints}—{untaints}



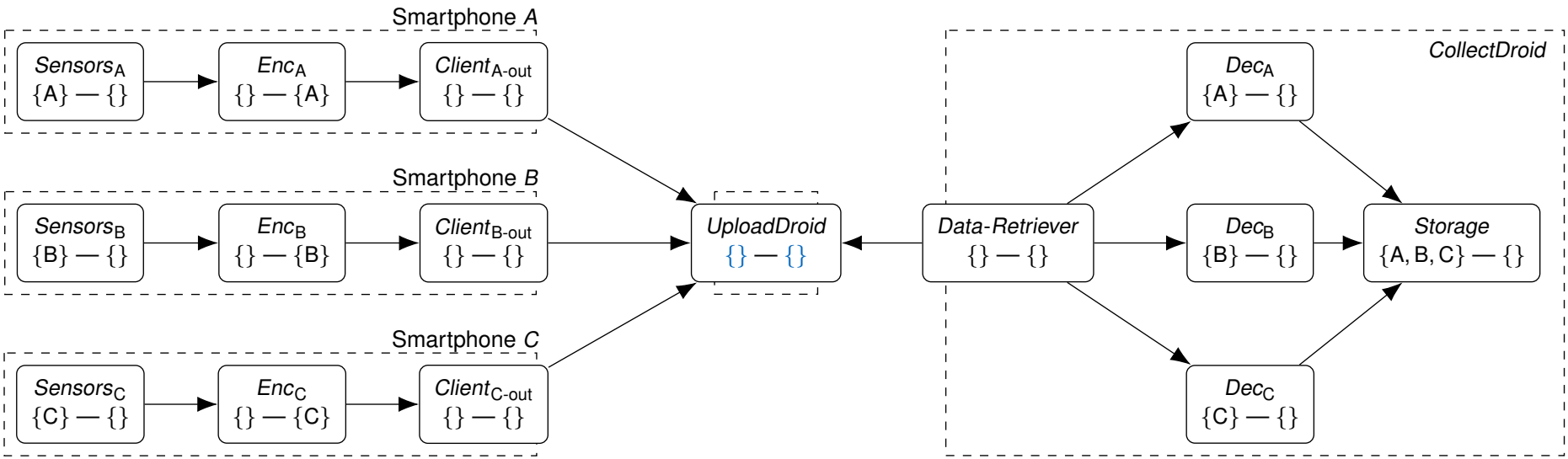
MeasrDroid – Architecture Formalized

- System for collecting smartphone sensor data
- Deployed at TUM
- Syntax: {taints}—{untaints}



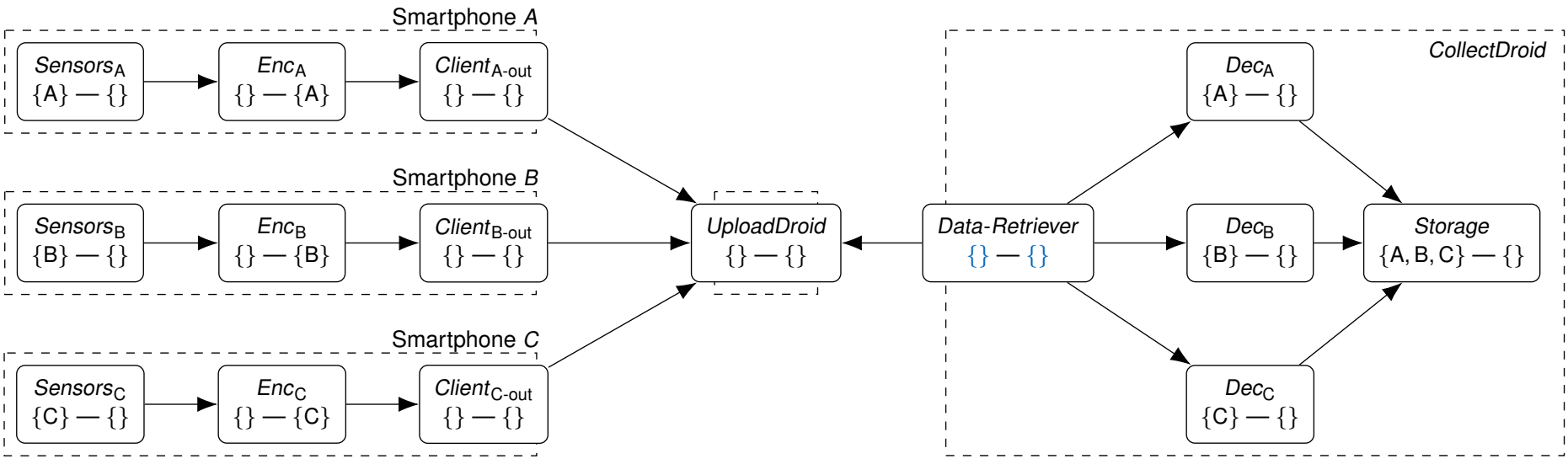
MeasrDroid – Architecture Formalized

- System for collecting smartphone sensor data
- Deployed at TUM
- Syntax: {taints}—{untaints}



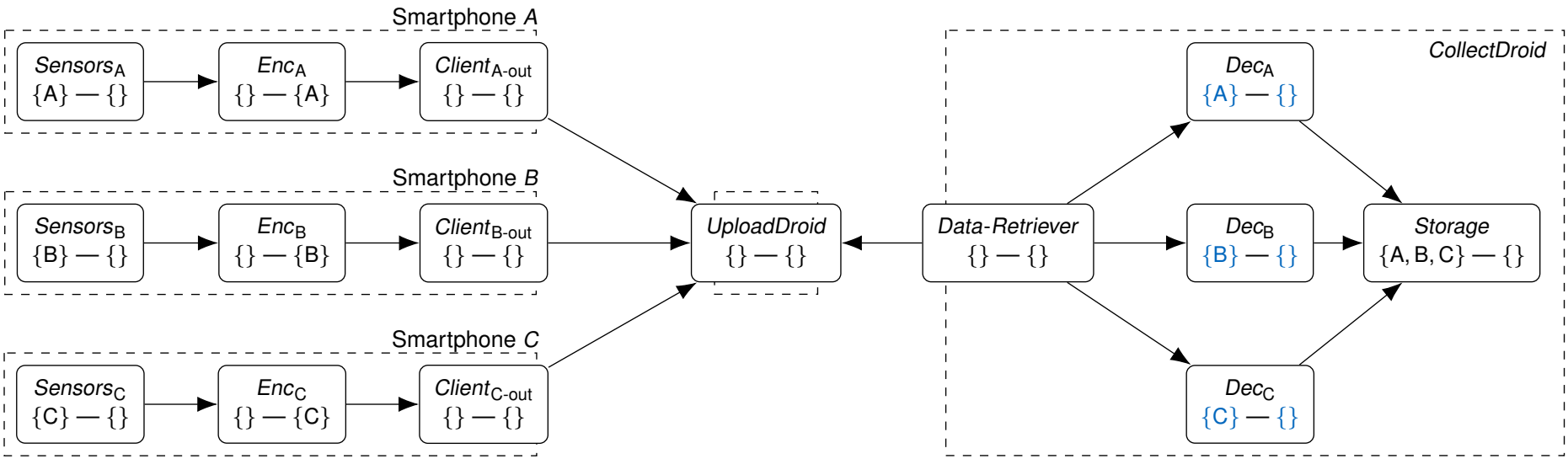
MeasrDroid – Architecture Formalized

- System for collecting smartphone sensor data
- Deployed at TUM
- Syntax: {taints}—{untaints}



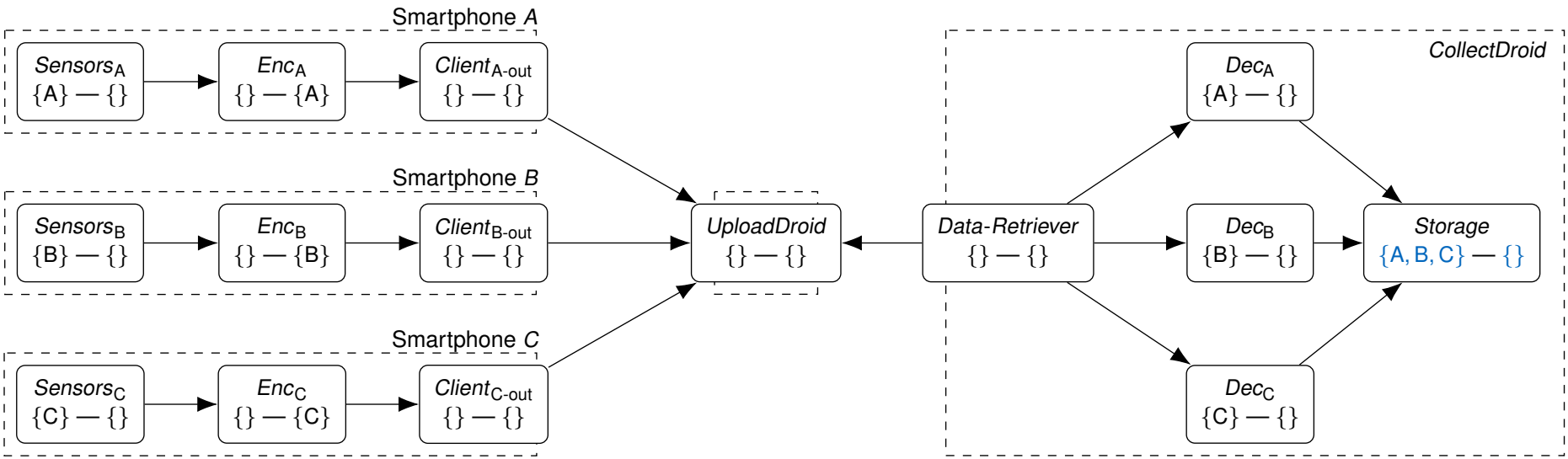
MeasrDroid – Architecture Formalized

- System for collecting smartphone sensor data
- Deployed at TUM
- Syntax: {taints}—{untaints}



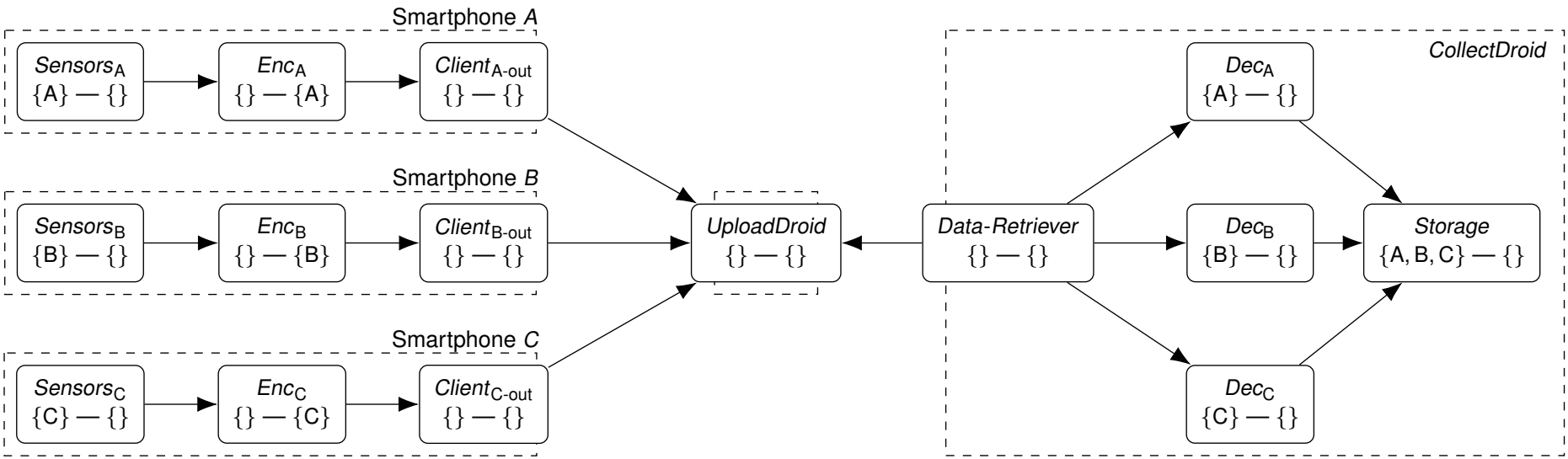
MeasrDroid – Architecture Formalized

- System for collecting smartphone sensor data
- Deployed at TUM
- Syntax: {taints}—{untaints}



MeasrDroid – Architecture Formalized

- System for collecting smartphone sensor data
- Deployed at TUM
- Syntax: {taints}—{untaints}



- Architecture valid according to taint analysis? ✓

MeasrDroid – Real-World Architecture Enforcement

- How is this architecture enforced?

MeasrDroid – Real-World Architecture Enforcement

- How is this architecture enforced?
- Central network firewall

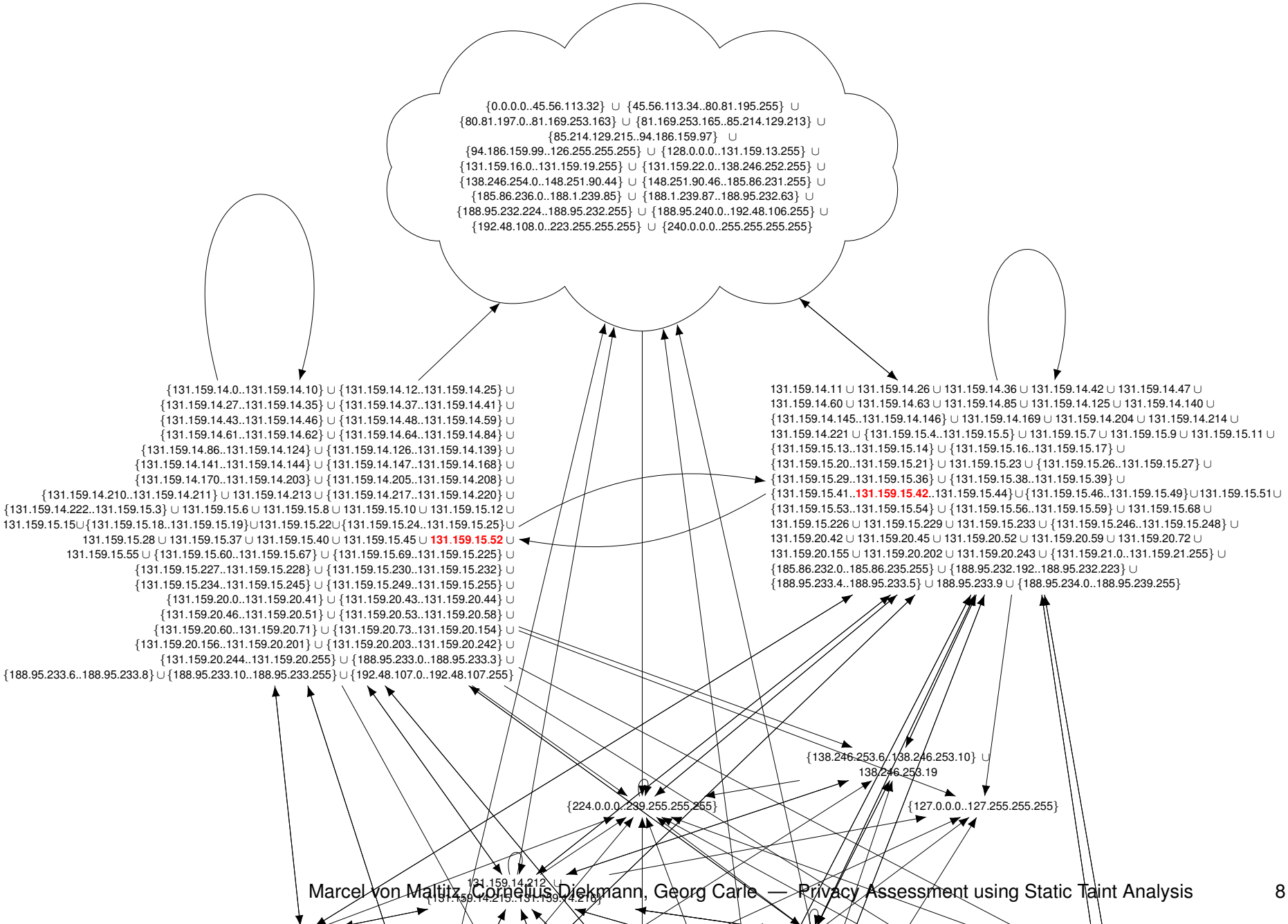
Get a graph they said

diekmann@xps12: ~/git/net-network/configs_chair_for_Network_Architectures_and_Services

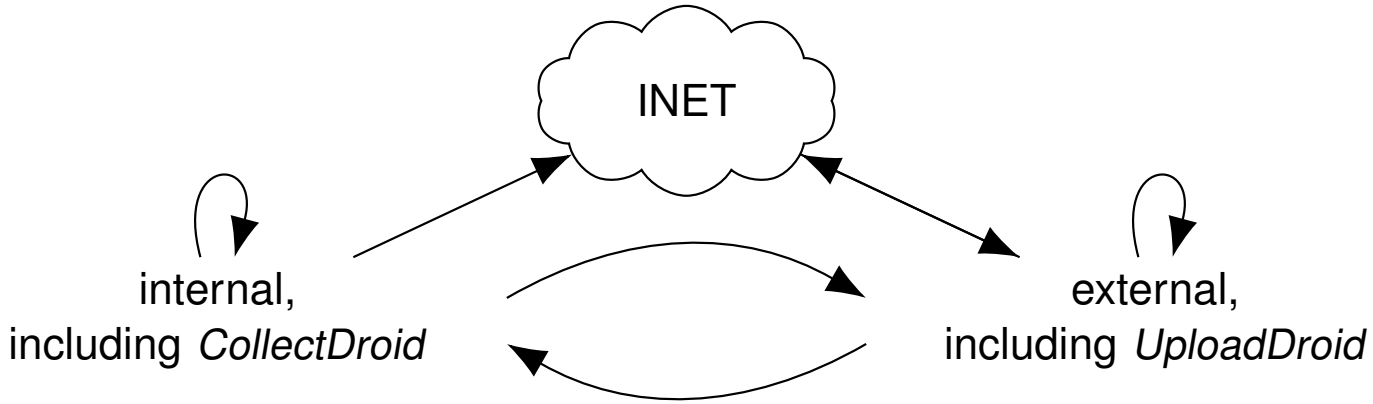
```
-A FORWARD -m state --state RELATED,ESTABLISHED,UNTRACKED -j ACCEPT
-A FORWARD -i eth1.110 -j NOTFROMHERE
-A FORWARD -i eth1.1024 -j NOTFROMHERE
-A FORWARD -m recent --update --seconds 60 --name DEFAULT --rsource -j LOG_RECENT_DROP2
-A FORWARD -p tcp -m state --state NEW -m tcp --dport 22 --tcp-flags FIN,SYN,RST,ACK SYN -
m recent --update --seconds 360 --hitcount 41 --name ratessh --rsource -j LOG_RECENT_DROP
-A FORWARD -s 127.0.0.0/8 -j LOG_DROP
-A FORWARD -s 131.159.14.221/32 -i eth1.1011 -j ACCEPT
-A FORWARD -s 131.159.15.252/32 -i eth1.152 -p udp -j ACCEPT
-A FORWARD -d 131.159.15.252/32 -o eth1.152 -p udp -m multiport --dports 4569,5000:65535 -
j ACCEPT
-A FORWARD -d 188.95.233.5/32 -i eth1.1011 -o eth1.97 -j ACCEPT
-A FORWARD -d 188.95.233.5/32 -o eth1.97 -p tcp -m tcp --sport 443 -j ACCEPT
-A FORWARD -s 131.159.15.247/32 -i eth1.152 -o eth1.110 -j ACCEPT
-A FORWARD -d 131.159.15.247/32 -i eth1.110 -o eth1.152 -j ACCEPT
-A FORWARD -s 131.159.15.248/32 -i eth1.152 -o eth1.110 -j ACCEPT
-A FORWARD -d 131.159.15.248/32 -i eth1.110 -o eth1.152 -j ACCEPT
-A FORWARD -d 131.159.14.26/32 -i eth1.110 -p tcp -m tcp --dport 22 -j ACCEPT
-A FORWARD -d 131.159.14.0/23 -i eth1.110 -p tcp -m state --state NEW -m tcp --dport 22 --
tcp-flags FIN,SYN,RST,ACK SYN -m recent --set --name ratessh --rsource
-A FORWARD -d 131.159.20.0/23 -i eth1.110 -p tcp -m state --state NEW -m tcp --dport 22 --
tcp-flags FIN,SYN,RST,ACK SYN -m recent --set --name ratessh --rsource
-A FORWARD -s 131.159.14.0/25 -i eth1.96 -j mac_96
-A FORWARD -i eth1.96 -j ranges_96
                                                              167,34        2%
```

It's easy they said

MeasrDroid – Real-World Architecture Enforcement

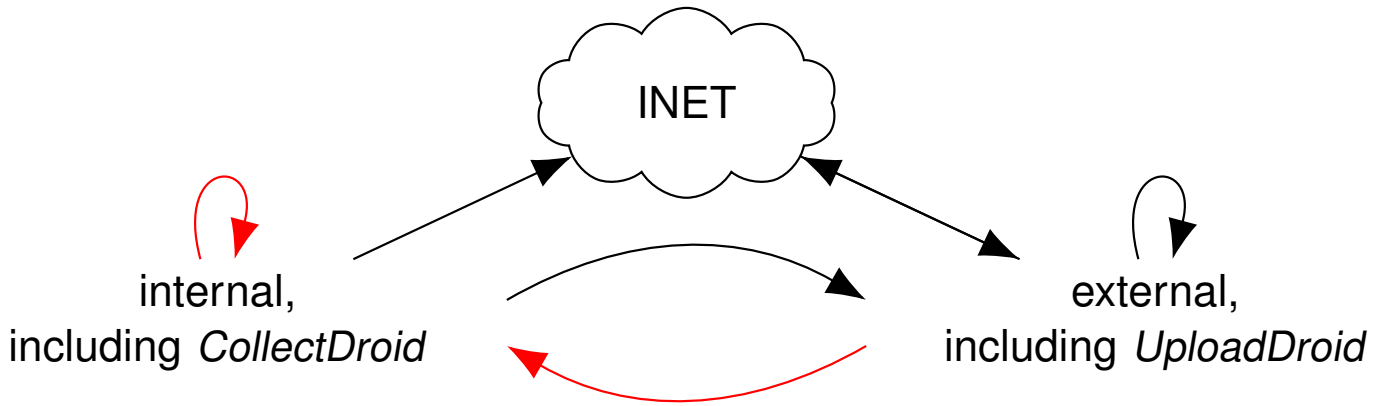


MeasrDroid – Real-World Architecture Enforcement



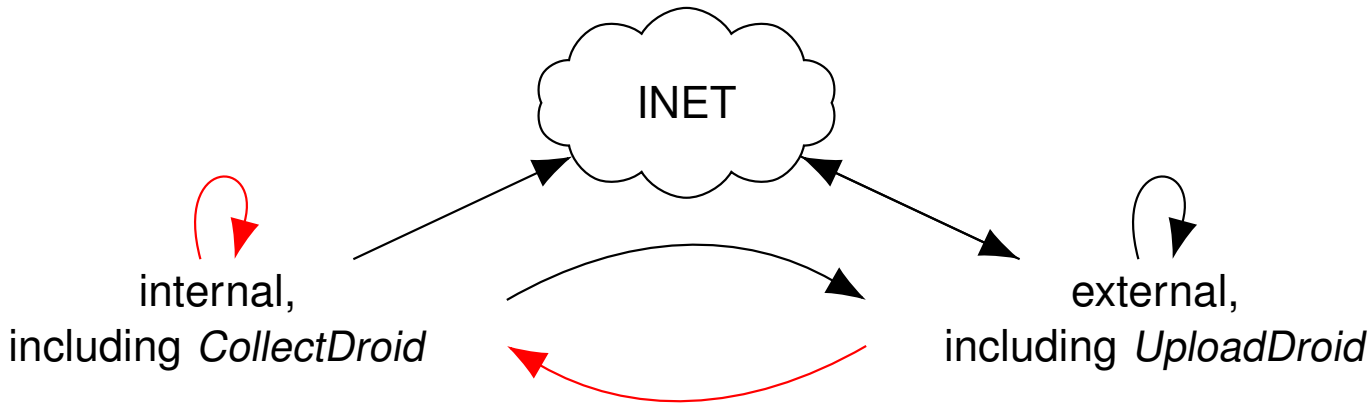
- Should be a graph isomorph to the architecture, disregarding
  - We cannot look inside the machines
  - All smartphones in INET

MeasrDroid – Real-World Architecture Enforcement



- Should be a graph isomorph to the architecture, disregarding
  - We cannot look inside the machines
  - All smartphones in INET
- Actually, two major architecture violations (red arrows)
  - *UploadDroid* can attack *CollectDroid*
  - Any compromised internal host can attack *CollectDroid*

MeasrDroid – Real-World Architecture Enforcement



- Should be a graph isomorph to the architecture, disregarding
  - We cannot look inside the machines
  - All smartphones in INET
- Actually, two major architecture violations (red arrows)
  - *UploadDroid* can attack *CollectDroid*
  - Any compromised internal host can attack *CollectDroid*
- Fixed by automated firewall generation feature of topoS

Conclusion

- Get your information flow as graph
- Add taint labels
- Do privacy analysis

Contributions

- Formalization of tainting, implemented in topoS
- Verified with Isabelle/HOL
- Two case studies

Tools

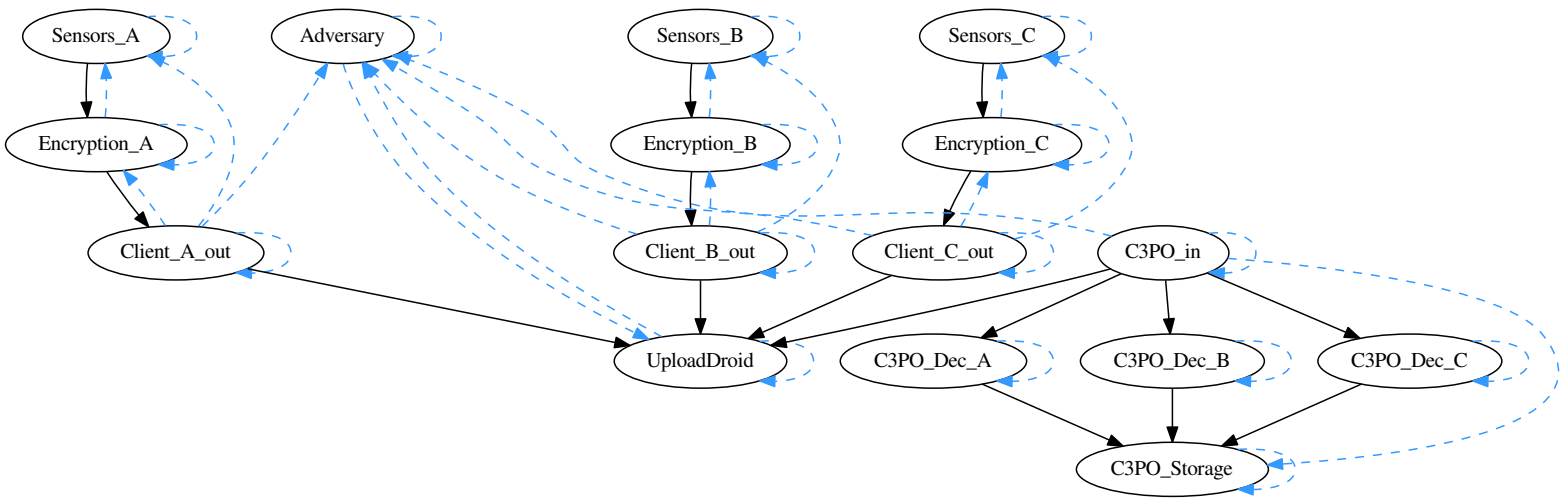
- C. Diekmann, A. Korsten, and G. Carle. Demonstrating topoS: Theorem-prover-based synthesis of secure network configurations. In *11th International Conference on Network and Service Management (CNSM)*, pages 366–371, Nov. 2015
- C. Diekmann, J. Michaelis, M. Haslbeck, and G. Carle. Verified iptables Firewall Analysis. In *IFIP Networking 2016*, Vienna, Austria, May 2016

- In the AFP & on github & `www.net.in.tum.de`

Backup Slides

Backup Slides

- How good is the architecture and requirement specification?



Screenshot of topoS

Backup Slides

Related Work: Dynamic Taint Analysis

- Taint analysis inspired by TaintDroid [9] and DroidDisintegrator [13]
- Dynamic runtime systems for individual Android devices/Apps
- Granularity: Android components

Bibliography

[1] Das Standard-Datenschutzmodell.
Technical report, Konferenz der unabhängigen Datenschutzbehörden des Bundes und der Länder, Darmstadt, 2015.

[2] K. Bock and M. Rost.
Privacy By Design und die Neuen Schutzziele.
DuD, 35(1):30–35, 2011.

[3] A. Cavoukian.
Creation of a Global Privacy Standard.
November 2006, Revised October 2009.

[4] A. Cavoukian.
Privacy by Design – The 7 Foundational Principles, Jan. 2011.

[5] G. Danezis, J. Domingo-Ferrer, M. Hansen, J.-H. Hoepman, D. L. Metayer, R. Tirtea, and S. Schiffner.
Privacy and Data Protection by Design - from policy to engineering.
Technical report, ENISA, 2015.

[6] C. Diekmann, A. Korsten, and G. Carle.
Demonstrating topoS: Theorem-prover-based synthesis of secure network configurations.
In *11th International Conference on Network and Service Management (CNSM)*, pages 366–371, Nov. 2015.

[7] C. Diekmann, J. Michaelis, M. Haslbeck, and G. Carle.
Verified iptables Firewall Analysis.
In *IFIP Networking 2016*, Vienna, Austria, May 2016.

[8] C. Diekmann, S.-A. Posselt, H. Niedermayer, H. Kinkelin, O. Hanka, and G. Carle.
Verifying Security Policies using Host Attributes.
In *FORTE*, pages 133–148, Berlin, Germany, June 2014. Springer.

Bibliography

[9] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth.
TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones.
ACM TOCS, 32(2):5, June 2014.

[10] J. Hoepman.
Privacy design strategies.
In *IFIP TC11 29th Int. Conf. on Information Security*, volume abs/1210.6621, 2012.

[11] M. Rost and A. Pfitzmann.
Datenschutz-Schutzziele – revisited.
Datenschutz und Datensicherheit DuD, 33(6):353–358, 2009.

[12] S. Steinbrecher and S. Köpsell.
Modelling Unlinkability.
Privacy Enhancing Technologies, 2760:32–47, 2003.

[13] E. Tromer and R. Schuster.
DroidDisintegrator: Intra-Application Information Flow Control in Android Apps (extended version).
In *ASIA CCS '16*, pages 401–412. ACM, 2016.