Cooperative Android App Analysis finally simple

Example Overview
Sources: \( s_1 \)  
Taint-Flows: \( s_1 \rightarrow s_4 \), \( s_1 \rightarrow s_7 \)  
Sinks: \( s_4 \), \( s_7 \)

To find both taint-flows (i) sources and sinks in Java and native code must be uncovered, (ii) reflection must be resolved and (iii) inter-app communication has to be analyzed.

Example AQL-Query
The simple query gets transformed into a more detailed one considering the challenges of the example.

Example AQL-Answer
The web representation shows that both expected taint-flows were identified by the cooperation of DroidRA, FlowDroid, IC3, NOAH and PIM

Usecases:
- ReproDroid: The Android Benchmark Reproduction Framework
- CoDiDroid: Cooperative (and Distributed) App Analysis Tool Framework
- Android Merge Tool → Benchmark Speed-Up & Analysis Lift-Up

AQL-Publications:
- 2017: Master's Thesis
- ESEC/FSE'18: Do Android Taint Analysis tools keep their promises? *
- ESEC/FSE'19: Together Strong: Cooperative Android App Analysis *
- ***'19: App Merging for Benchmark Speed-Up and Analysis Lift-Up

* Artifacts evaluated / under review
- Src-Code
- Executables
- Tutorials
- Benchmark Extensions
- Evaluation Results
- ...

Felix Pauck  
ipauck@mail.uni-Paderborn.de  
http://FelixPauck.de  
https://FoelliX.github.io