

# SONOMA: Secure Opportunistic Network-Wide Offload for Mobile Applications

Aaron Gember & Aditya Akella, *University of Wisconsin - Madison*

## BACKGROUND

 Offloading migrates part of a running mobile application and executes it on a secondary resource (desktop, server, etc.).

### Why are existing systems not widely adopted?

- 1) No mechanisms are in place to maintain the *security* and *privacy* of data used by an offload application.
- 2) The focus is on *what* to offload instead of *where* to offload, and they assume dedicated local resources are available for offloading.

**SONOMA** is our mobile application offloading system designed to *opportunistically* leverage available resources and offer *security guarantees* and *performance/energy improvements* to smartphones.

## SECURITY

### What information is at risk?

Data accessed or generated by an application becomes part of its runtime state and may be included in the state transferred during offloading. We consider three types of data:

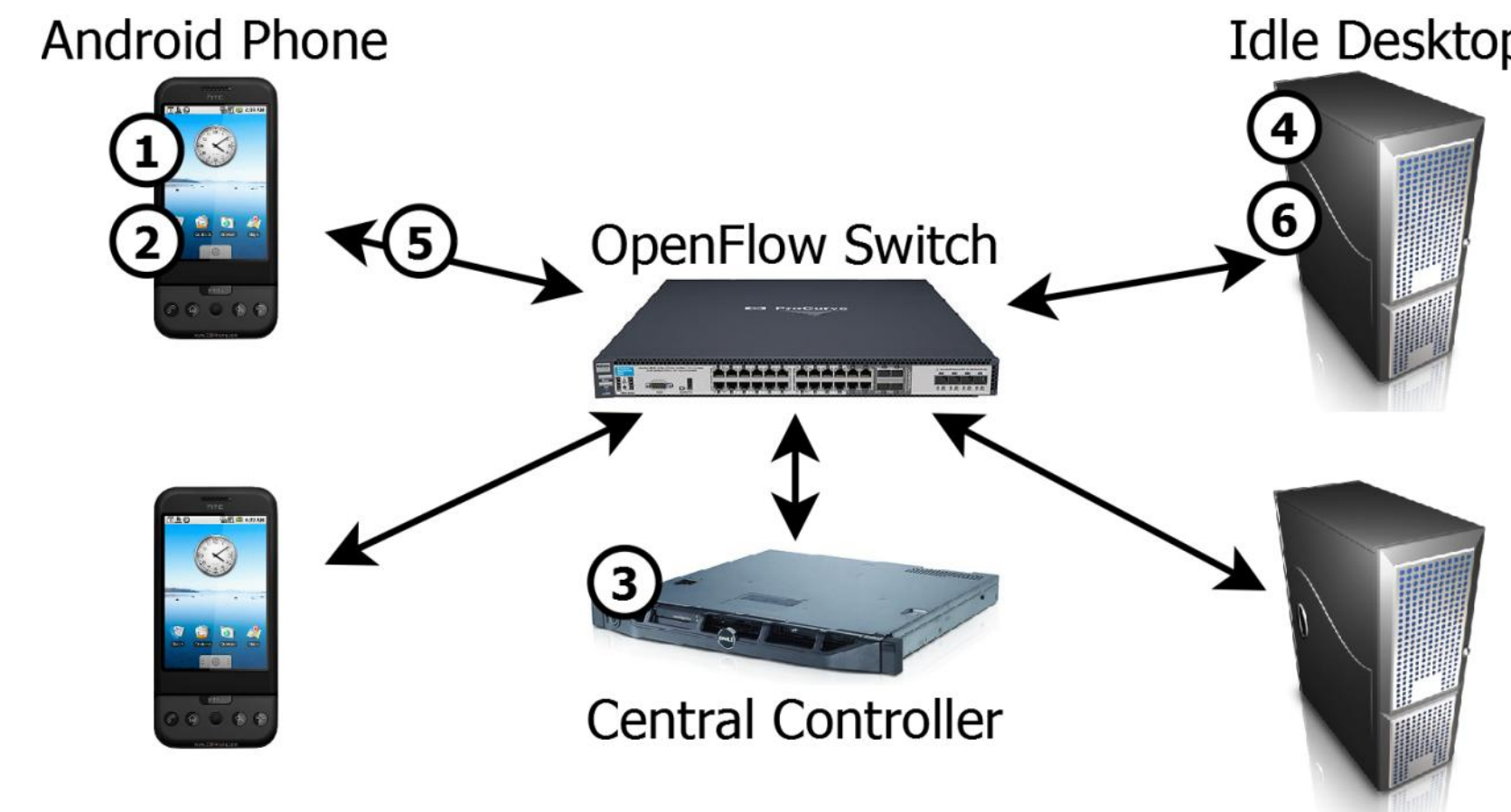
- 1) *No-private*: data raises no privacy or security concerns.
- 2) *Enterprise-private*: should not be leaked outside the enterprise.
- 3) *User-private*: data should be accessed only by the current user.

### What does SONOMA do to ensure security?

A security policy on the central controller defines which resources provide the necessary trust for *user-* and *enterprise-private* data. The decision of *where* to offload limits the resources considered based on this policy. Also, communication will be encrypted for *user-private* data, but not for the other data types to avoid undue overhead.



## DEMONSTRATION



UW - Madison OpenFlow Network

- 1) Launch compute intensive application on *Android Phone*
  - 2) Offload point reached; application asks phone agent to request resources from the central controller
  - 3) Controller pairs available resource with offload request
  - 4) Application connects to desktop agent and sends package name; desktop agent launches new Dalvik VM and listens for state
  - 5) Phone transfers execution state to new VM instance
  - 6) Desktop executes code and returns result; desktop VM is killed
- Multiple phones and desktops perform this process simultaneously.

## RESOURCE SCHEDULING

Measurements show there are significant idle computational resources in a typical enterprise. SONOMA pairs these resources with offloading requests to provide the *most benefit* to as *many* mobile devices as possible. Every time an application reaches an offload point, the mobile device asks the central controller *where* to offload. The controller assigns the application to a desktop, server, or cloud with sufficient resources, optionally throttling back already offloaded applications to provide some benefit to the mobile device.