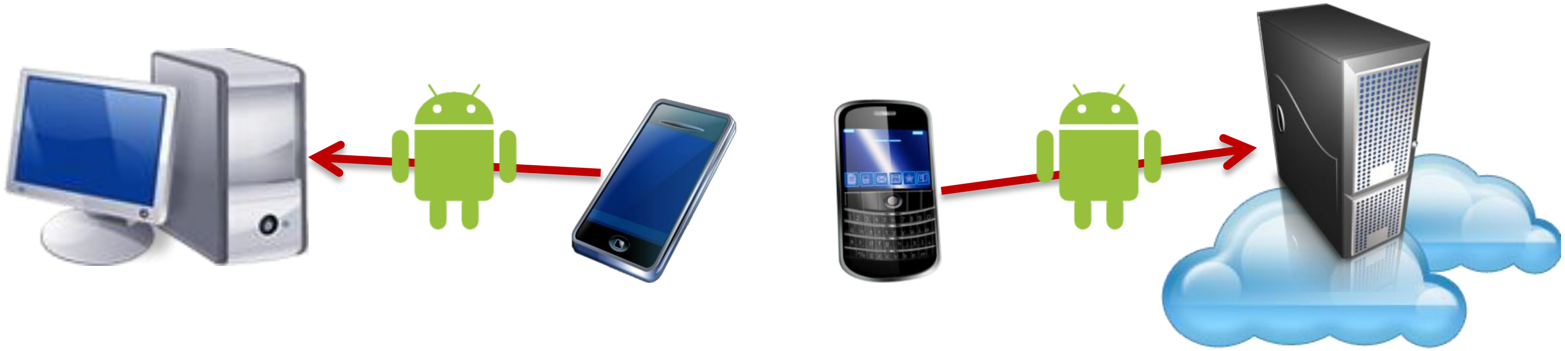


Enterprise Centric Offload System

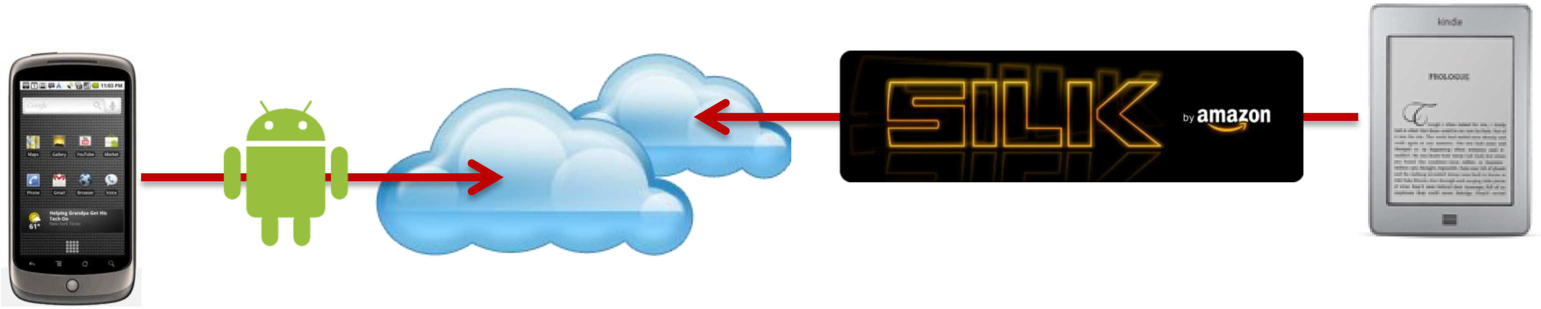
Aaron Gember & Aditya Akella

UNIVERSITY OF WISCONSIN–MADISON

Coping with Handheld Demands

- Increasingly complex mobile applications
  - High CPU and energy demands – e.g., speech recognition
  - Work with sensitive data – e.g., health records

- Application-independent offloading
  - Enables performance improvements and energy savings without re-writing applications
  - E.g., CloneCloud [*Chun ‘11*], MAUI [*Cuervo ’10*], Silk



Roadblocks to Offloading

Privacy and trust: sensitive data in application state can be leaked in transit or on resources

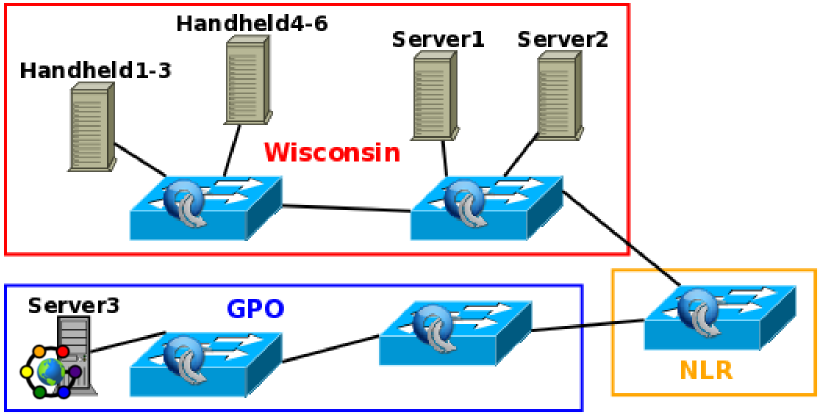
Resource availability: diverse set of compute resources with varying availability and capacity

Scalability: multiple handhelds with different goals must be able to simultaneously offload

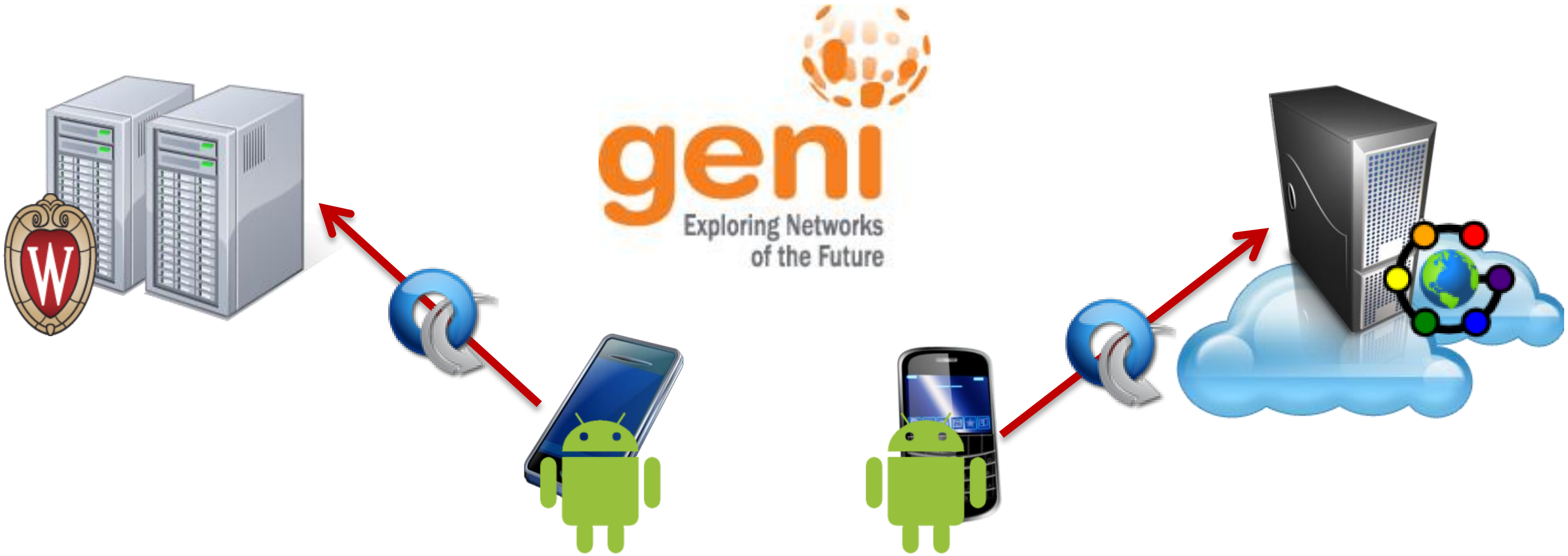
Eliminate these roadblocks to make offloading feasible for enterprise networks

Experimentation in geni

- **How do we assign offloads to compute resources to provide the most benefits to many handheld users?**
  - Varying goals – latency improvement, energy savings
  - Limited set of trusted resources
  - Changing resource availability, and diverse capabilities
  - Overhead of state transfer – state size, channel quality



- GENI Resources
  - protogeni
  - OpenFlow



Live Demo

How our Enterprise Centric Offloading System (ECOS) assigns compute resources