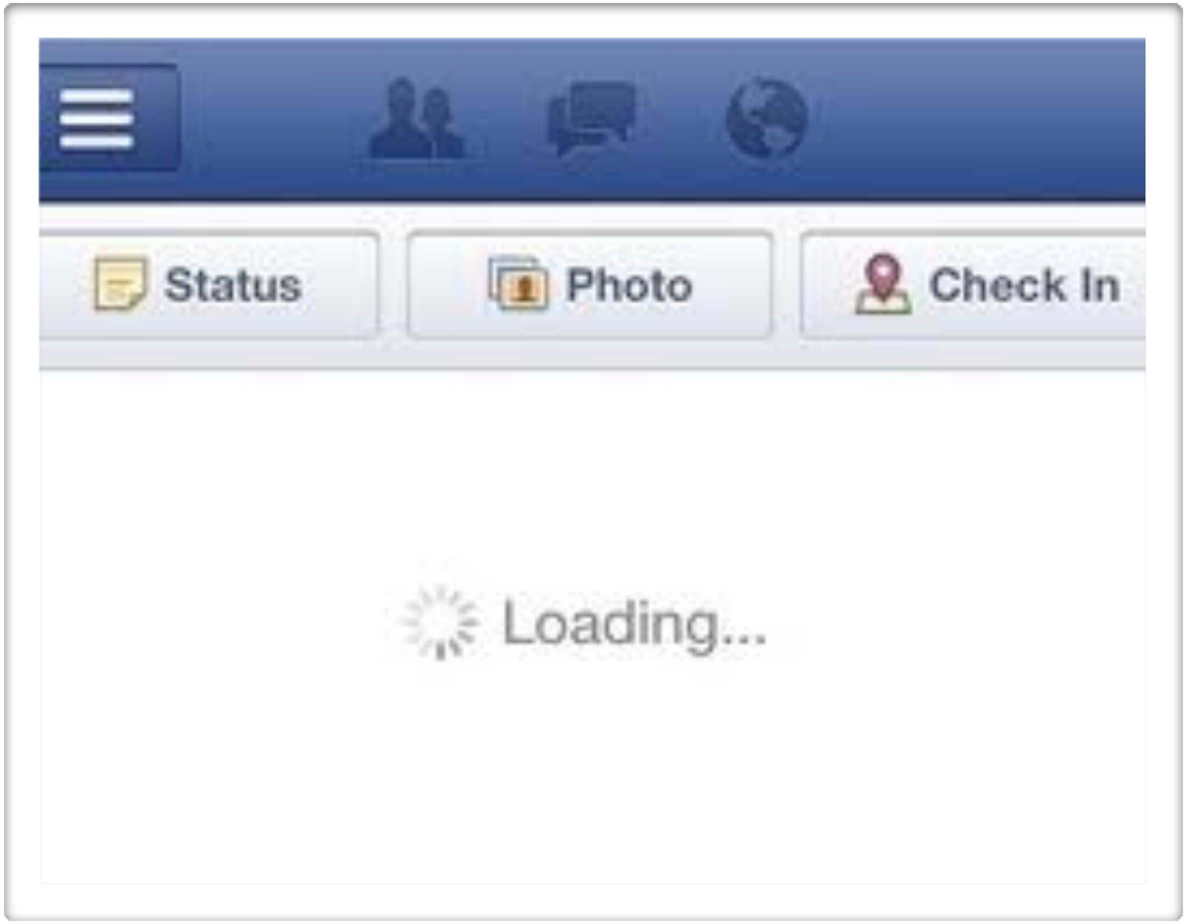
Towards an Open Mobile Measurement Platform

David Choffnes

University of Washington

Along with University of Michigan and Google

Mobile Internet can be terrible



Mobile monitoring can help us do better

Goals: Check up on carrier performance, predictive comparison shopping, ...

- Requires us to understand application-perceived performance
  - **Where** the device is used
  - **When** the device is used

Key challenge: Multiple apps have the same/similar goals in mind, want to pervasively monitor the network

A strawman proposal

Let the market decide which app will get the best coverage

- Mobiperf?
- MySpeedTest?
- SamKnows?
- Tempo?
- Netalyzer-droid?

Every app for itself won’t work

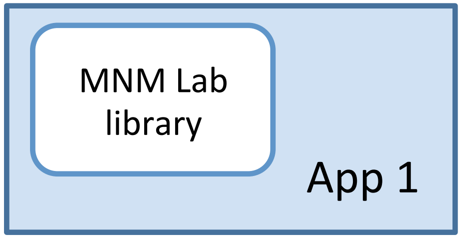
- Adoption/barrier to entry
  - Does any one of us really have a killer app?
- Interference
  - Conflicting/synchronous measurements
- Measurement validation
  - Grad student code!
- Data management
  - Collecting, storing, publishing traces

A Sustainable Proposal

One *measurement library* to rule them all

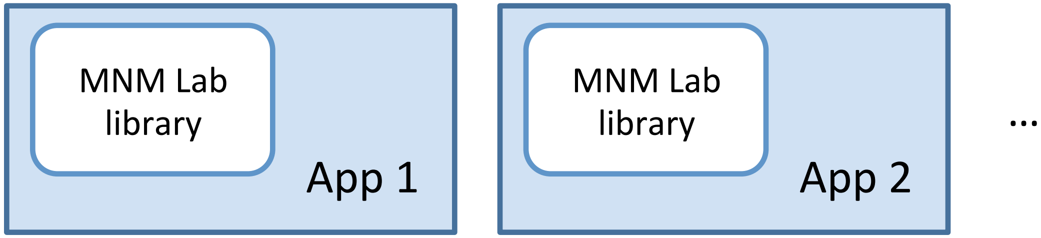
A Sustainable Proposal

One *measurement library* to rule them all



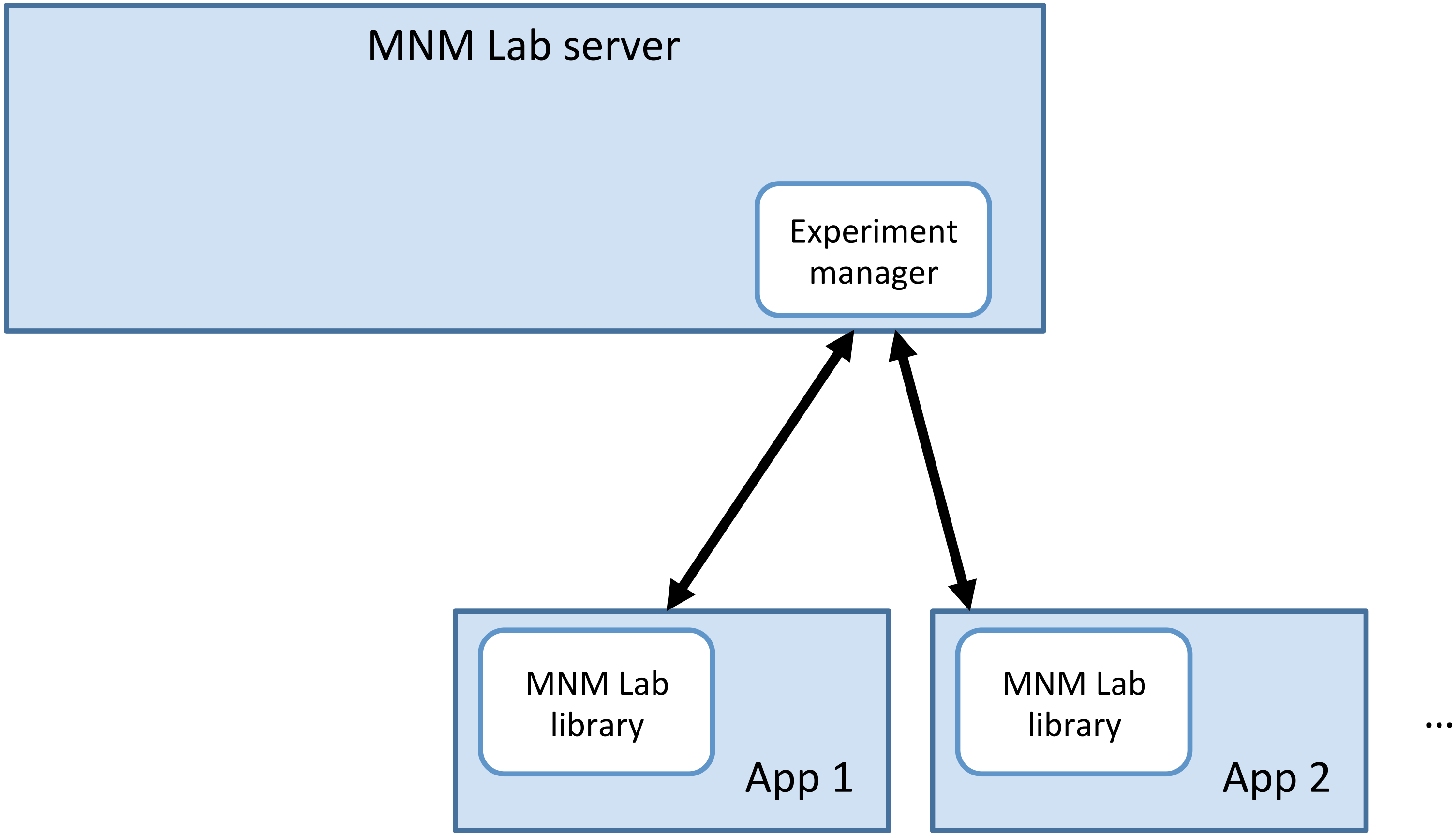
A Sustainable Proposal

One *measurement library* to rule them all



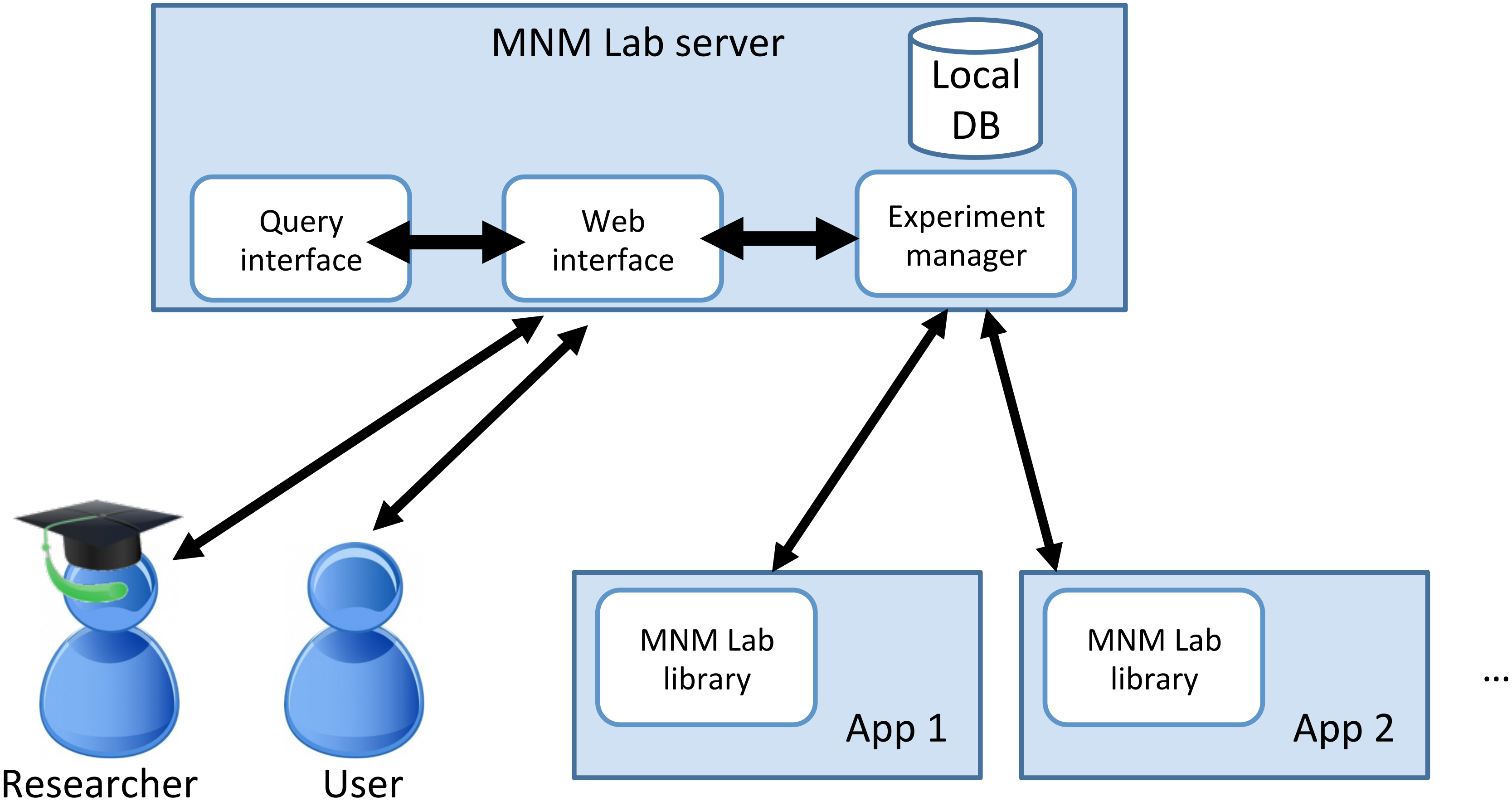
A Sustainable Proposal

One *measurement library* to rule them all



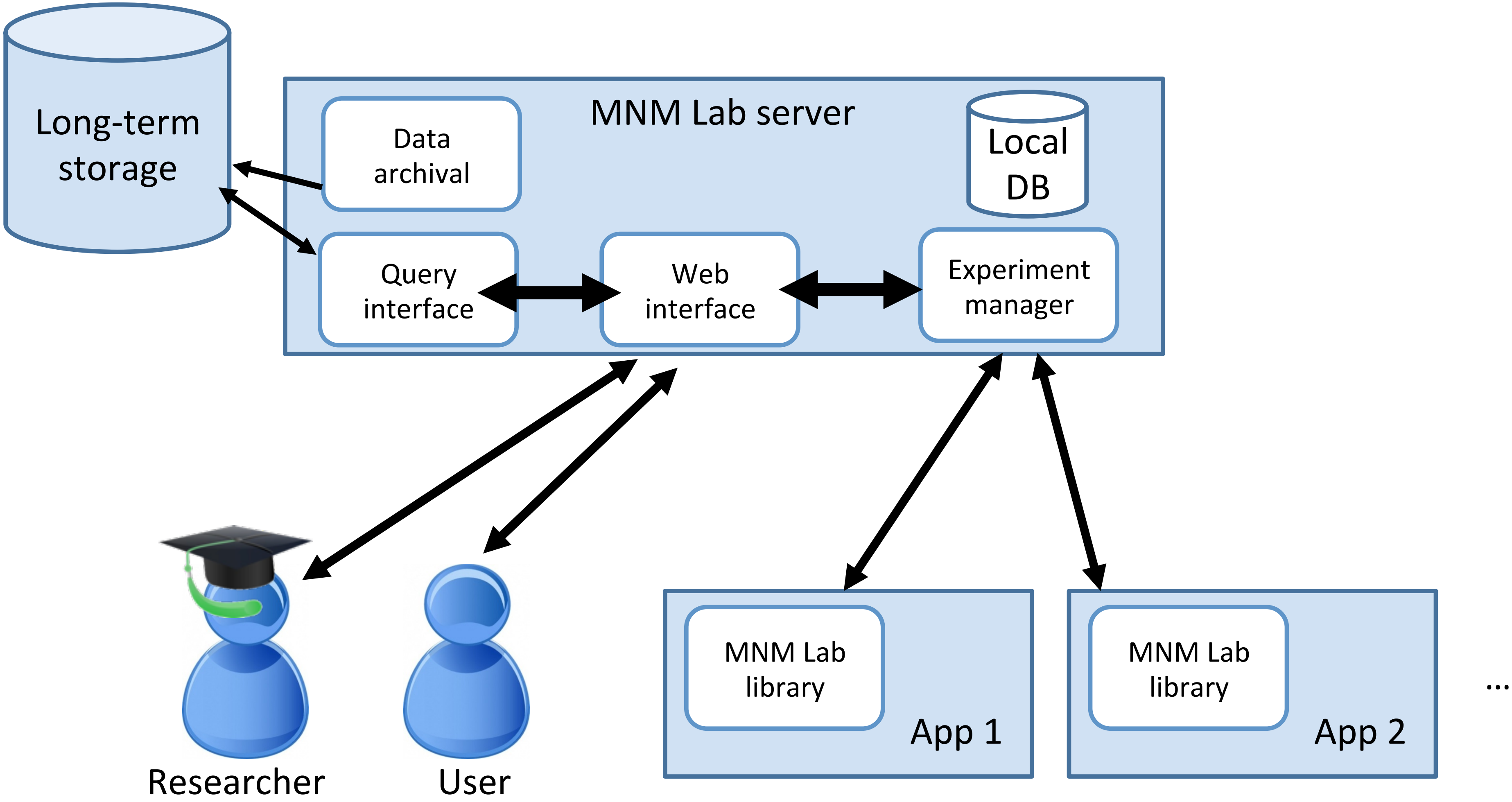
A Sustainable Proposal

One *measurement library* to rule them all



A Sustainable Proposal

One *measurement library* to rule them all



Key advantages

- **Adoption**:
  Easy to integrate into *any* app (Angry Birds?)
  Go forth and make popular apps!
- **Validation**:
  Write-once-use-everywhere validated measurement primitives
- **Management**:
  AppEngine server (h/t Google) provides management/ collection of measurement experiments
- **Data sharing**:
  Continuous feed to Google Storage public bucket
- **Incentives/Open Access**:
  Researchers get quota in proportion to library-enabled app instances they bring to the system (like RIPE Atlas)

A few tricky problems

- Managing user resources efficiently
  - Be smart about when to measure, when not to
  - Don't suck up too much battery, data quota
- Interface for programmers
  - Dasu? (Declarative programming)
  - Other ideas?
- Curated experiments
  - Prevent abuse
  - Allow reuse of existing modules

Current status

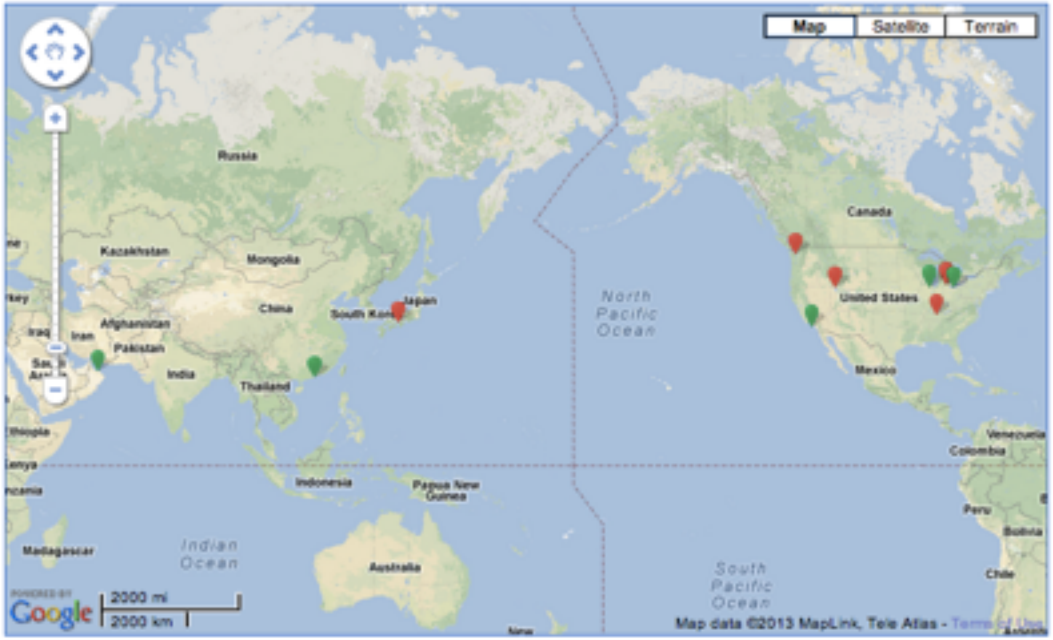
- Mobiperf to become “reference app” for library
  - Primitive measurement scheduler
  - Data already being stripped of PII and published daily

- Dominic Hamon (M-Lab) is building native-code measurement primitives for a library

- Working with 802.16.3 (mobile measurement standard)

Obligatory answers to 3 questions

- Data sharing: gs://openmobiledata_public
- Visualization: http://openmobiledata.appspot.com



- What I want to get out of this:
  Users, measurement primitives, experiments needed