Immunization Cold Chain Status Reporting via SMS

Jenny Kang, Isaac Reynolds,

Jackson Roberts & Nicholas Shahan

Cold Chain Failures

- Failures in vaccine distribution and storage equipment lead to waste and shortages
- Existing reporting tools are slow, difficult to use, and inaccurate
- Our system will make reporting fast and easy to improve reliability, accuracy, and efficiency

Health Ministries Need More Data

- Effective distribution of vaccines requires a well-coordinated cold chain
- Managers of vaccine inventories will benefit from timely reporting of inventory and cold chain failures
- Data reporters will benefit from easier reporting

Current Work in Laos

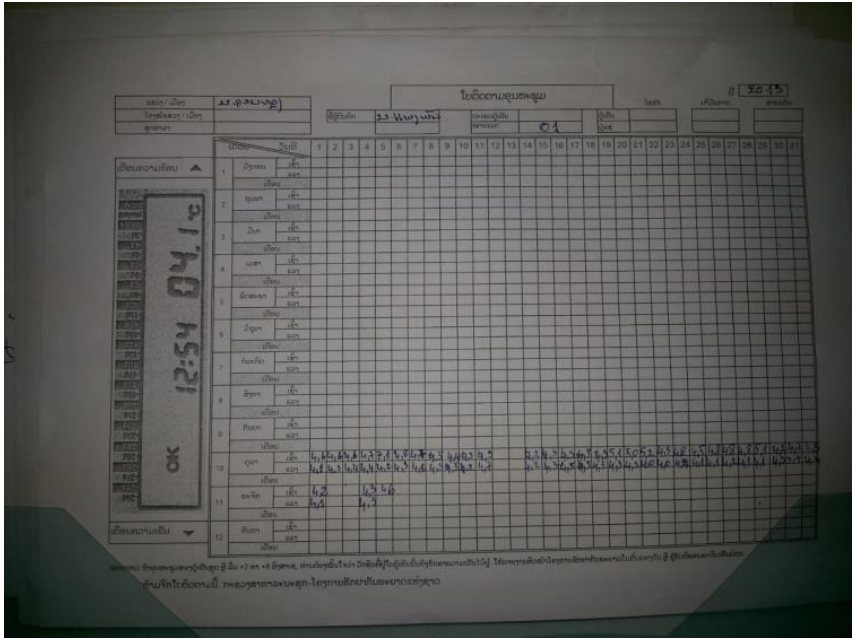
- Richard Anderson, Fahad Pervaiz, and Trevor Perrier are developing a prototype to deploy in Laos
- Prototype is limited by time frame and incomplete development of a CCEI (fridge data) module for DHIS2
  - New development guided by CCEM (MS Access)

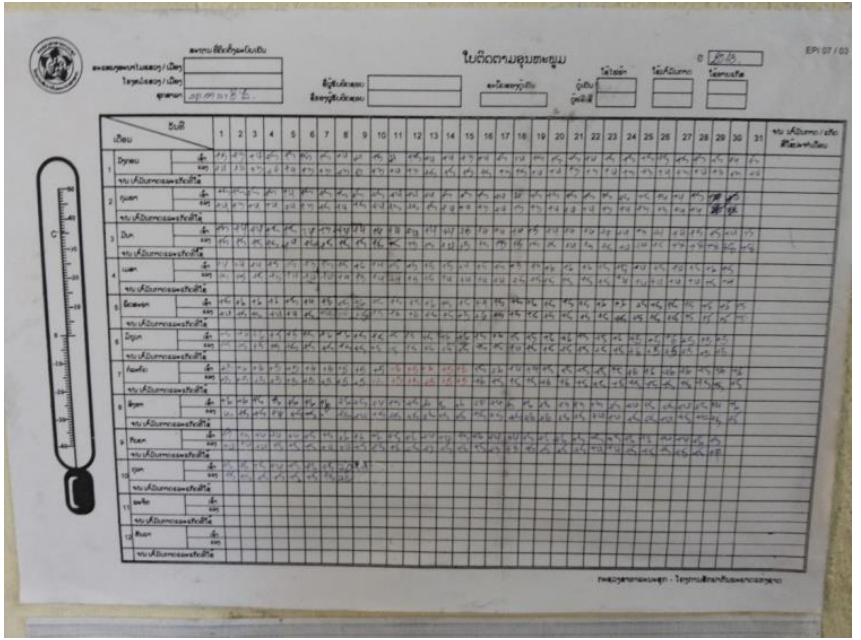
Province store

Refrigerators were labeled “A” , “B”, “C”… for easy identification in SMS reporting (example from province and district store)

Demonstration of using fridge tag

Staff demonstrated good understanding of reading and using data on fridge tags.



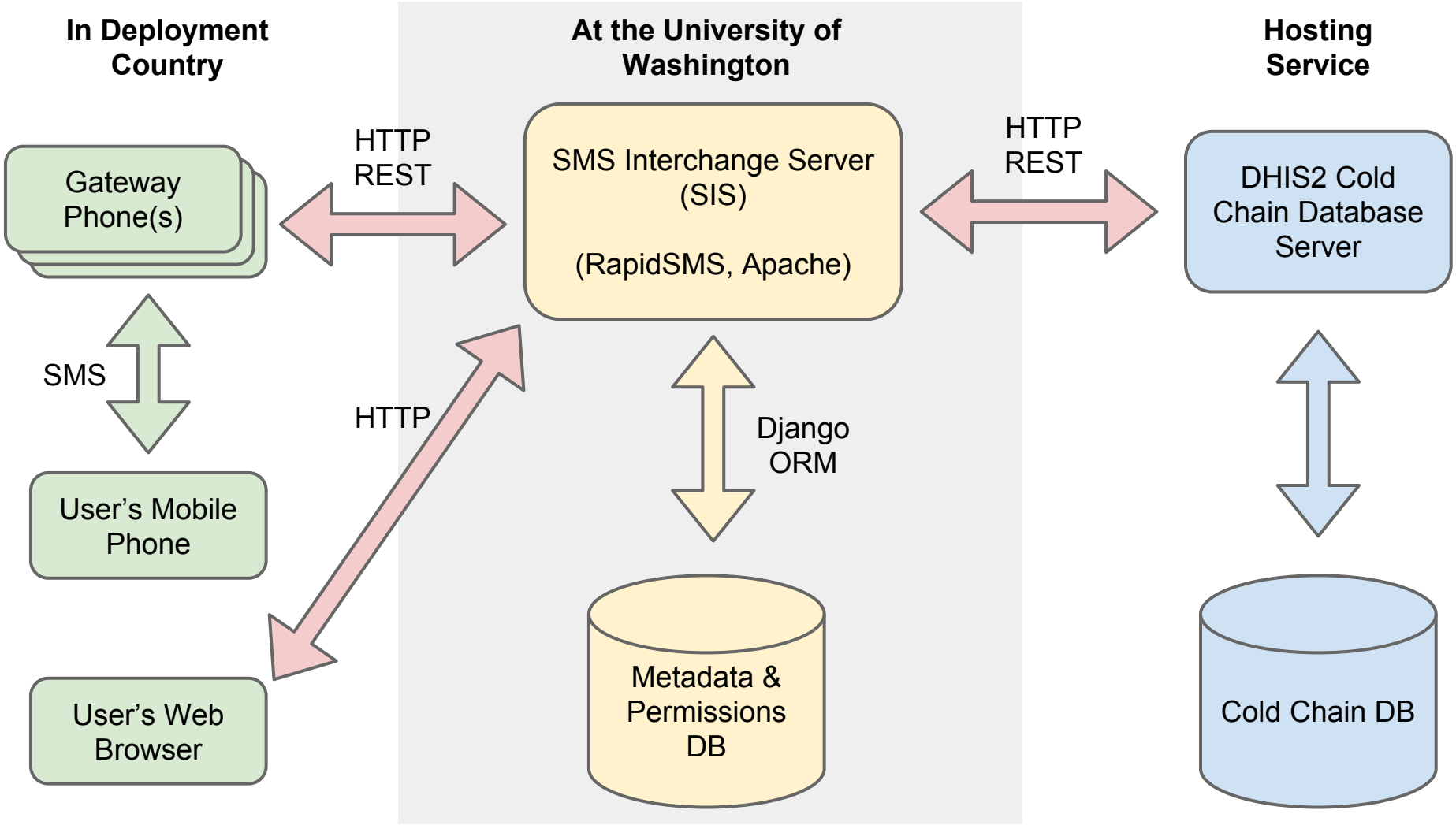


Monthly record of daily temperature

Partially implemented forms to record temperature from previous month.
Currently distributing forms to provinces in Laos.

Project Description

- Build an SMS Interchange Server (SIS) that accepts SMS messages and
  - Sends and queries data to and from DHIS2
  - Sends notifications in response to events such as fridge failures
  - Manages users and associated metadata
  - Provides a web-based administration interfaces


In Deployment Country
At the University of Washington
Hosting Service
Gateway Phone(s)
HTTP REST
SMS Interchange Server (SIS)
(RapidSMS, Apache)
HTTP REST
DHIS2 Cold Chain Database Server
SMS
HTTP
Django ORM
User's Mobile Phone
User's Web Browser
Metadata & Permissions DB
Cold Chain DB

Design Challenges

- Representing DHIS2 entities in SIS
- Generalizing the existing prototype
- Representing permissions and notifications/ subscriptions
- Communicating with non-technical stakeholders to derive project requirements

Design & Evaluation

- We will compare our implementation to the prototype that Fahad and Trevor are building
  - Implement their external specification
  - Avoid their mistakes
  - Address the obstacles they face
  - Duplicate their successes
- We will continue to meet with UNICEF to discuss progress and future requirements

Plan for Next Quarter

- Infrastructure + Design: 2 weeks
- Implementation + Evaluation: 6 weeks
  - Fortnightly deliverable milestones
- Finishing touches and coursework: 2 weeks

Plan for Next Quarter

Week:

1. Requirements; design.
2. **Finish design.** System setup; education.
3. Basic SMS I/O; testing framework.
4. **Finish SMS API prototype.** Simple SMS APIs; users.
5. All SMS APIs; permissions/roles.
6. **Finish SMS API, web interface prototype.**
7. Subscriptions/notifications; web interface.
8. **Deliver software.** Web interface.
9. Project report, presentation, poster, video, etc.
10. **Finish coursework.**

Long-term improvements

- Monitor DHIS2 to detect changes that bypassed SIS and send notifications based on these changes