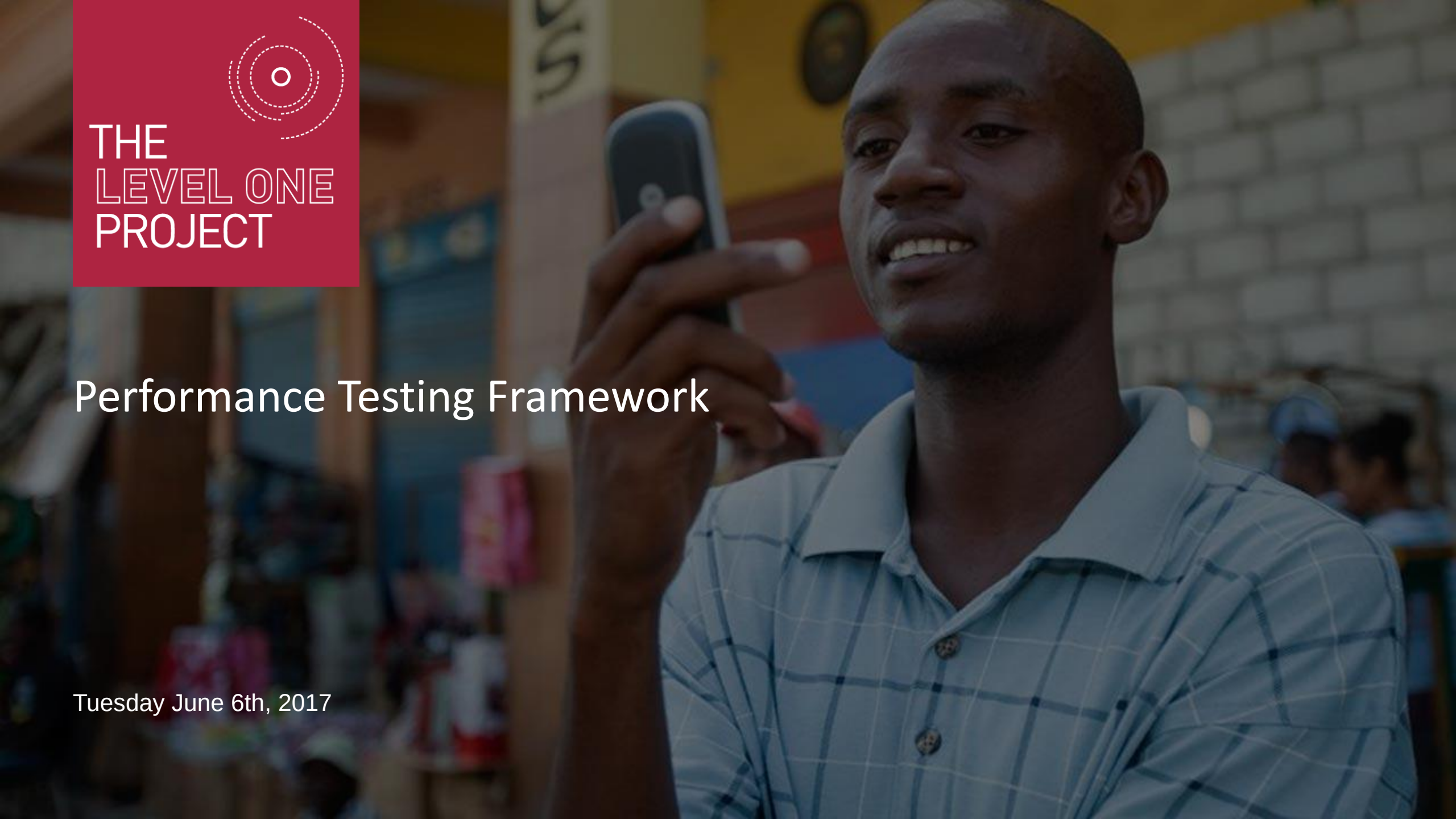
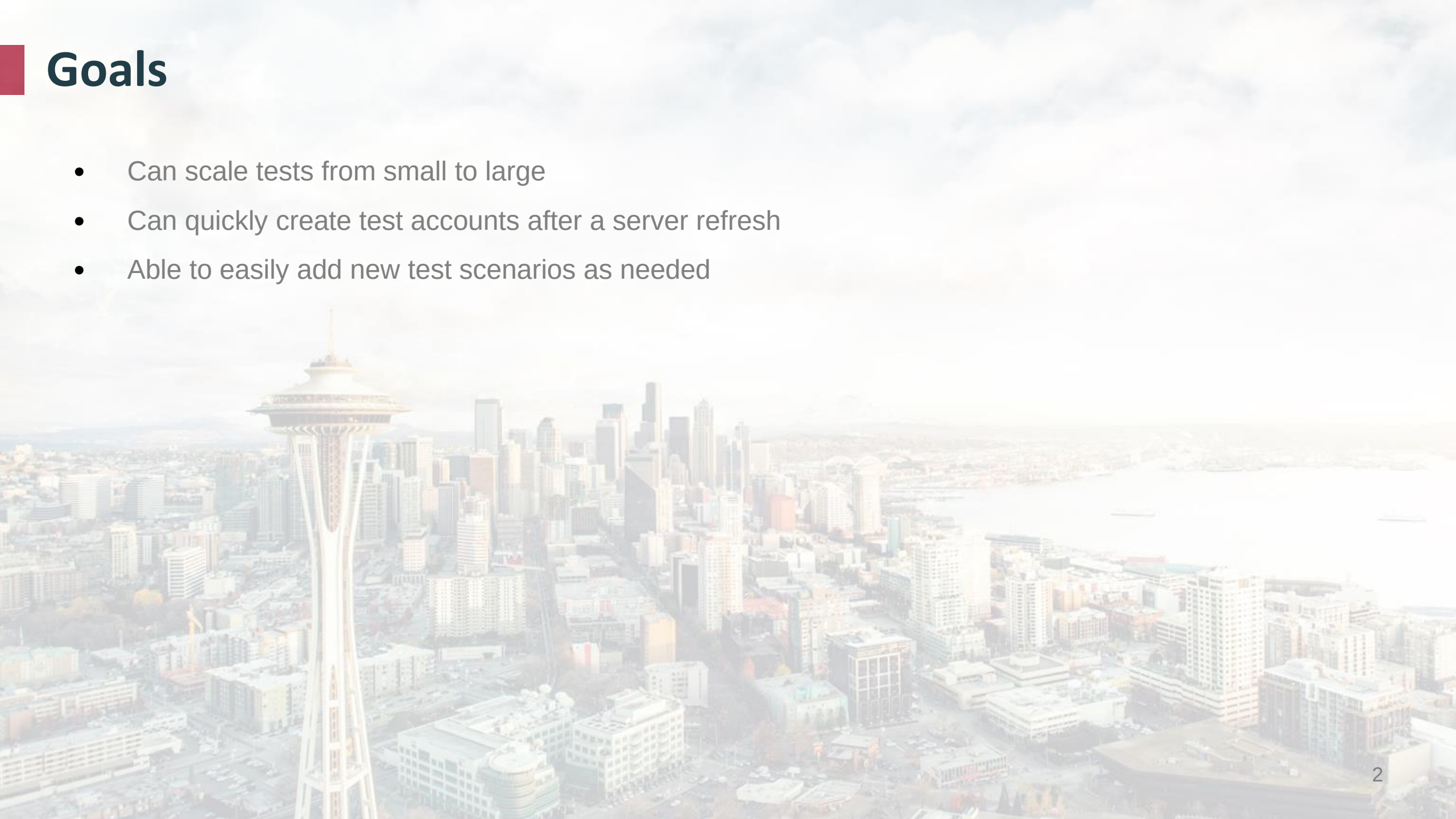


Performance Testing Framework

Tuesday June 6th, 2017

Goals

- Can scale tests from small to large
- Can quickly create test accounts after a server refresh
- Able to easily add new test scenarios as needed



What are the performance tools

Tools

- JMeter – Runs test scenarios (steps/tasks)
- User account creation service – Creates new set of accounts for USSD tests
- Metrics – Records performance behavior statistics

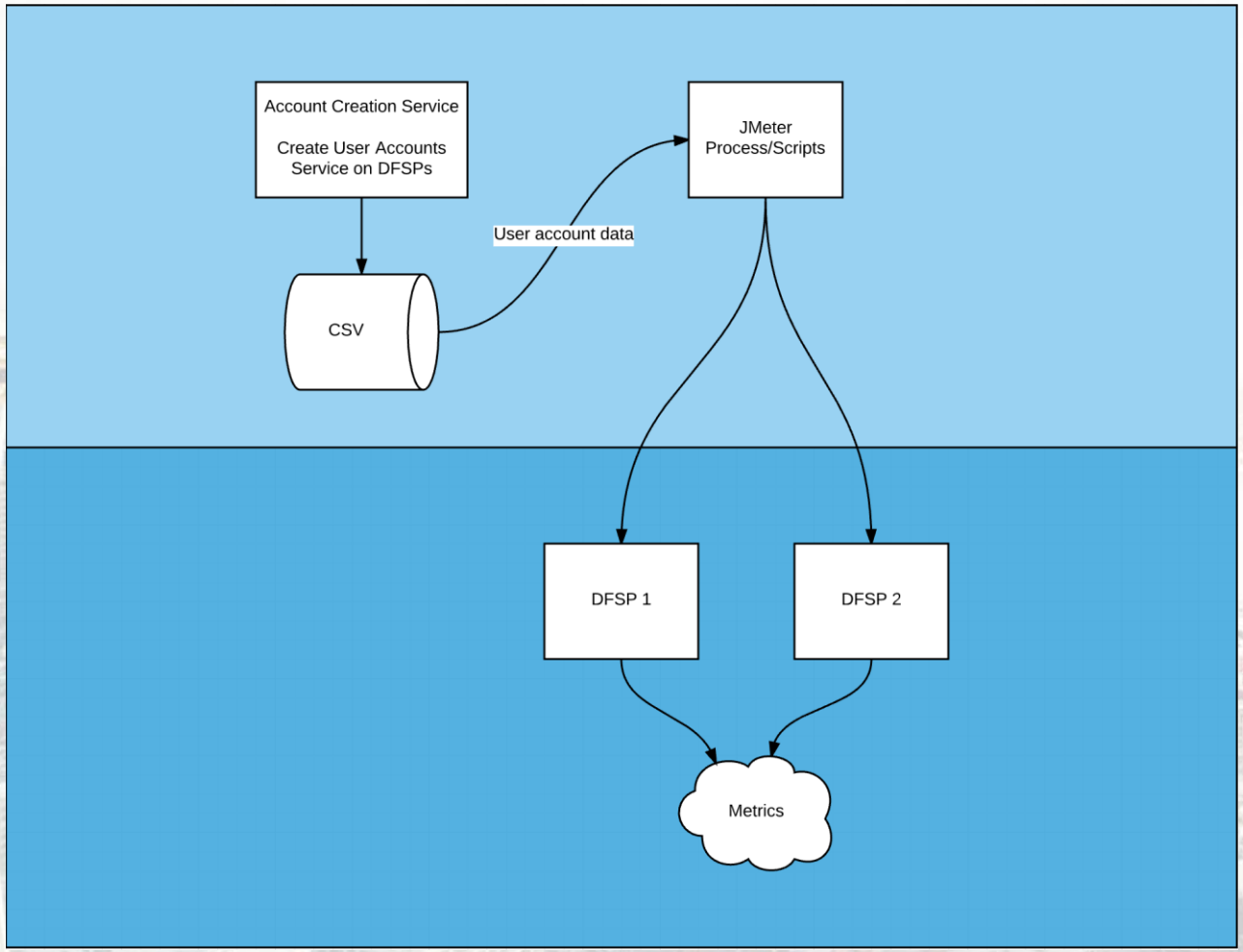
For USSD

- Send money
- Sell goods

For Interop Services

- Performance test ILP Ledger Adapter using mocks (for initial testing)

Overview


Account Creation Service
Create User Accounts Service on DFSPs
JMeter Process/Scripts
User account data
CSV
DFSP 1
DFSP 2
Metrics

Performance testing approach/methodology

Develop testing template (next PI)

- This will help ensure tests are run a consistent and predictable manner

Using template:

- Start with a fresh performance test environment
- Create user data on each DFSP (1&2)
  - # of users to meet this goal of the test, i.e. 100
  - Done through service, create csv used by JMeter
- Launch JMeter test
- Gather and record key statistic data points
  - Observe Metrics UI data points

Can use this approach to compare results on different AWS configurations

Performance testing environment

Performance environment

- AWS M4Large
  - 2 CPU
  - 8 GB RAM
  - 450 Mbps dedicated EBS bandwidth

Interop standalone environment

- AWS M4Large
  - 2 CPU
  - 8 GB RAM
  - 450 Mbps dedicated EBS bandwidth

Metrics data

All the interop services captures several data points of metrics

There are several types of metrics

- Counters
- Timers
- Rates

Counters

- Count -- Last reported value of count
- Interval Count -- Change in count value between reporting intervals

Timers

- Mean, Min, Median, Max, Std Dev, 75, 95, 98, 99, 999th Percentiles

Rates

- 1, 5, 15 minute rates -- Exponentially weighted moving average

Thank you