Scarlet SmallTalk

John McIntosh
johnmci@smalltalkconsulting.com

Michael Rueger
michael@andience.co.nz

"~~Amber~~ Scarlet is written in itself,
including the compiler,
and compiles into efficient JavaScript."

–www.amber-lang.net

A BIT OF BACKGROUND…

LABWARE

- LabWare LIMS
  Laboratory Information Management System

- Countless industries world wide

- 100.000 daily users

- 1.000+ modules

LIMS

- Implemented in VSE Smalltalk
- Multi-MB of code
- No feasible way to run Smalltalk on device
- Loadable Modules

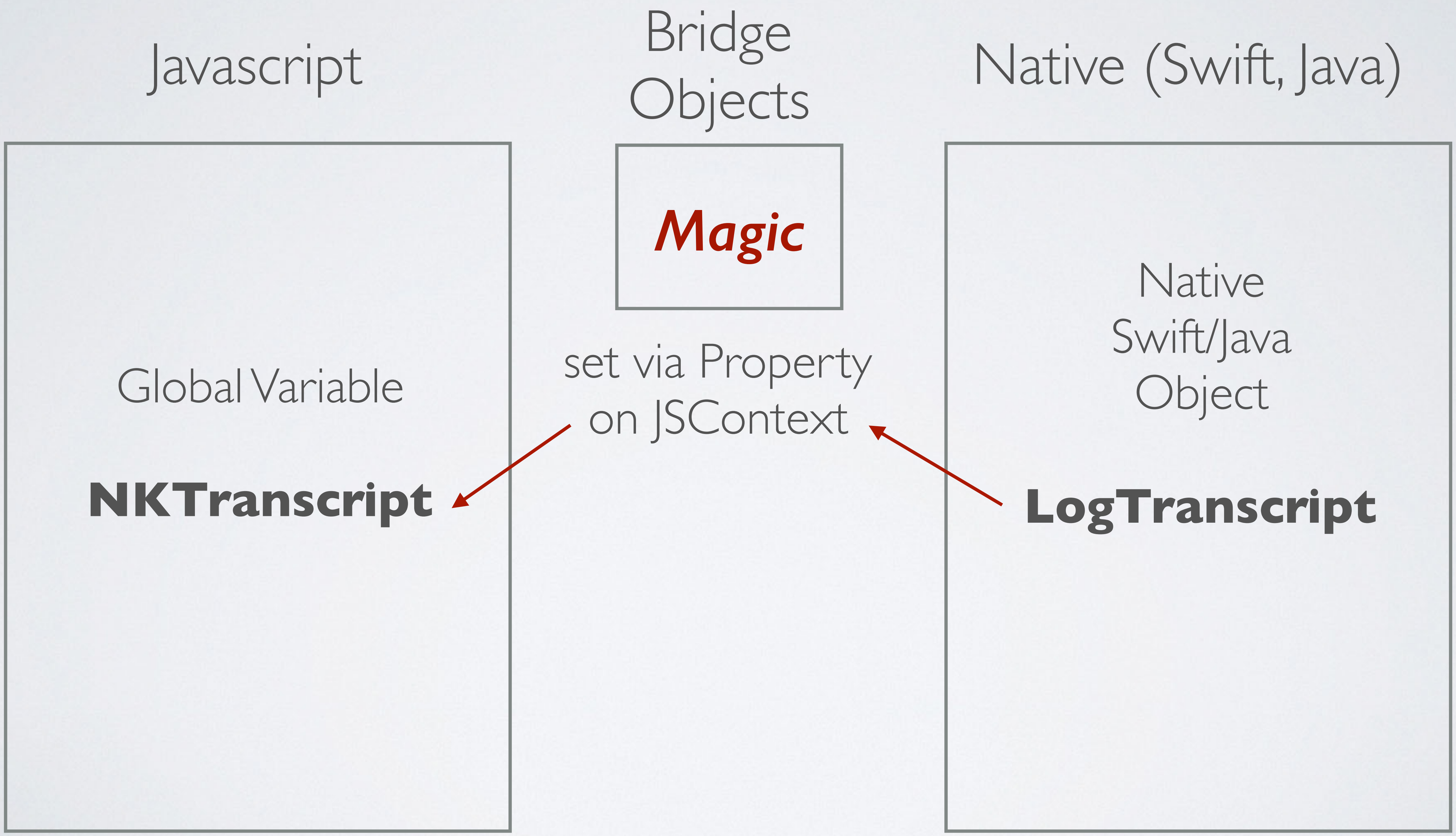
LIMS ON MOBILE

- Decision to cross-compile to Javascript
- Add functionality to integrate device capabilities
  - UI Components
  - Camera
  - Map

JAVASCRIPT INTEGRATION

- Javascript - Native Bridge
  - iOS
    Apple Javascript Core, Swift
  - Android
    Custom Javascript Core port, Java/JNI

JAVASCRIPT INTEGRATION



JAVASCRIPT (SMALLTALK) RUNTIME

S8

- Based on Amber
  (before it was called Amber)
  (Yes, we are aware of the history)
- Uses an outdated Javascript VM
- Geared towards use in browser
- High memory usage

S8

- Uses inline Javascript
  -> not back portable into Smalltalk
- Compiler based on (old) PetitParser
- No useful compiler error messages
- Unreadable code
- Very slow compilation of large files

SCARLET

- Compiler based on Squeak Compiler
- Code generation strongly influenced by modern Amber (from a year ago)
- Proper compiler error messages
- Readable code
- Fast compilation, linear time
  20-40 times faster than S8
- Linear memory usage

SCARLET

- Faster Runtime (30-50%)
- Primitives instead of inline Javascript
  Introduce a small overhead
- No inline Javascript
  - Compiler developed in Squeak
  - Running in Squeak or Scarlet
  - Also ported to VSE

INLINE JAVASCRIPT
VS
PRIMITIVES

INLINE JAVASCRIPT

```
Transcript

nextPutAll: aString
{' console.log(aString) '}.
```

PRIMITIVE INVOCATION

Smalltalk method with standard primitive annotation:

```
Transcript

nextPutAll: aString
<primitive: 'primNextPutAll' module: 'SKTranscript'>
```

INVOCATION TRANSLATED TO JAVASCRIPT

```
function Transcript_nextPutAll_(aString) {

  var $$primResult = SKTranscript.primNextPutAll(this,
arguments);

  if ($$primResult !== primFailValue) {
    return $$primResult;
  }
  self.primitiveFailed();
}
```

PRIMITIVE IMPLEMENTATION CONSOLE MODE

```
SKTranscript.primNextPutAll = function (receiver,
args) {
    var aString = args[0];
    if (typeof aString !== 'string') {
        return this.primFailValue;
    }
    console.log(aString);
}
```

PRIMITIVE IMPLEMENTATION MOBILE DEVICE

```
SKTranscript.primNextPutAll = function (receiver,
args) {
  var aString = args[0];
  if (typeof aString !== 'string') {
    return this.primFailValue;
  }
  NKTranscript.nextPutAll(aString);
}
```

native (Swift,Java) method

Native Object (Swift bridge, Java JNI)

SCARLET COMMAND LINE

SCARLET COMMAND LINE

```
./scarlet
Usage: scarlet [options] [command] <files-to-load...>

Options:
  -i, --interactive    Interactive mode
  -h, --help           output usage information

Commands:
  compile <source...>  Compile a file or a directory of files
  build <source>       Compile files in a directory into an image
```

INTERACTIVE MODE

```
./scarlet -i
> 3+4
Result: 7

> 3 squared
Result: 9

> (1 to: 10) collect: [:i | i squared]
Result: 1,4,9,16,25,36,49,64,81,100

> Transcript show: 'hello world'
hello world
Result: {st:Transcript}
```

COMPILE/ BUILD

Build a custom Scarlet image (`scarlet build example`):

- Source files
  ```
  example/NumberFunctions.st
  example/Prompter.st
  ```
- Translated Javascript files
  ```
  example/NumberFunctions.st.js
  example/Prompter.st.js
  ```
- Combined with Scarlet kernel image
  ```
  example/mobile.js
  ```

SCARLET
MOBILE INTEGRATION

SCARLET MOBILE INTEGRATION

- Setup up JSCore context

```
jscContext = new JscContext(this);
```

- Load custom Scarlet image

```
jscContext.evaluateScript("mobile.js");
```

- Set the property for the JS-native bridge object/variable

```
jscContext.property(
      "NKTranscript",
      new LogTranscript(jscContext));
```

Java Transcript implementation

- Load JS Transcript primitive

```
jscContext.evaluateScript("SKTranscript.js");
```

SCARLET MOBILE INTEGRATION

- Invoking *Transcript show:* from Java

```
jscContext.stEvaluateSync(
   "Transcript show: 'hello world from smalltalk'");
```

- Implementation of the Java Transcript primitive function

```
public void nextPutAll(String message) {
    Log.d("transcript", message);
}
```

- Log output on Android

```
2019-08-21 14:16:51.201 3149-3149/
org.javascriptcore.android.example D/transcript: hello world
from smalltalk
```

…AFTER APPLYING A LOT MORE MAGIC…

LABWARE MOBILE

- 20+ MB of Smalltalk code translated to Javascript
- 100+ native primitive/bridge functions
- 2000+ JIRA entries

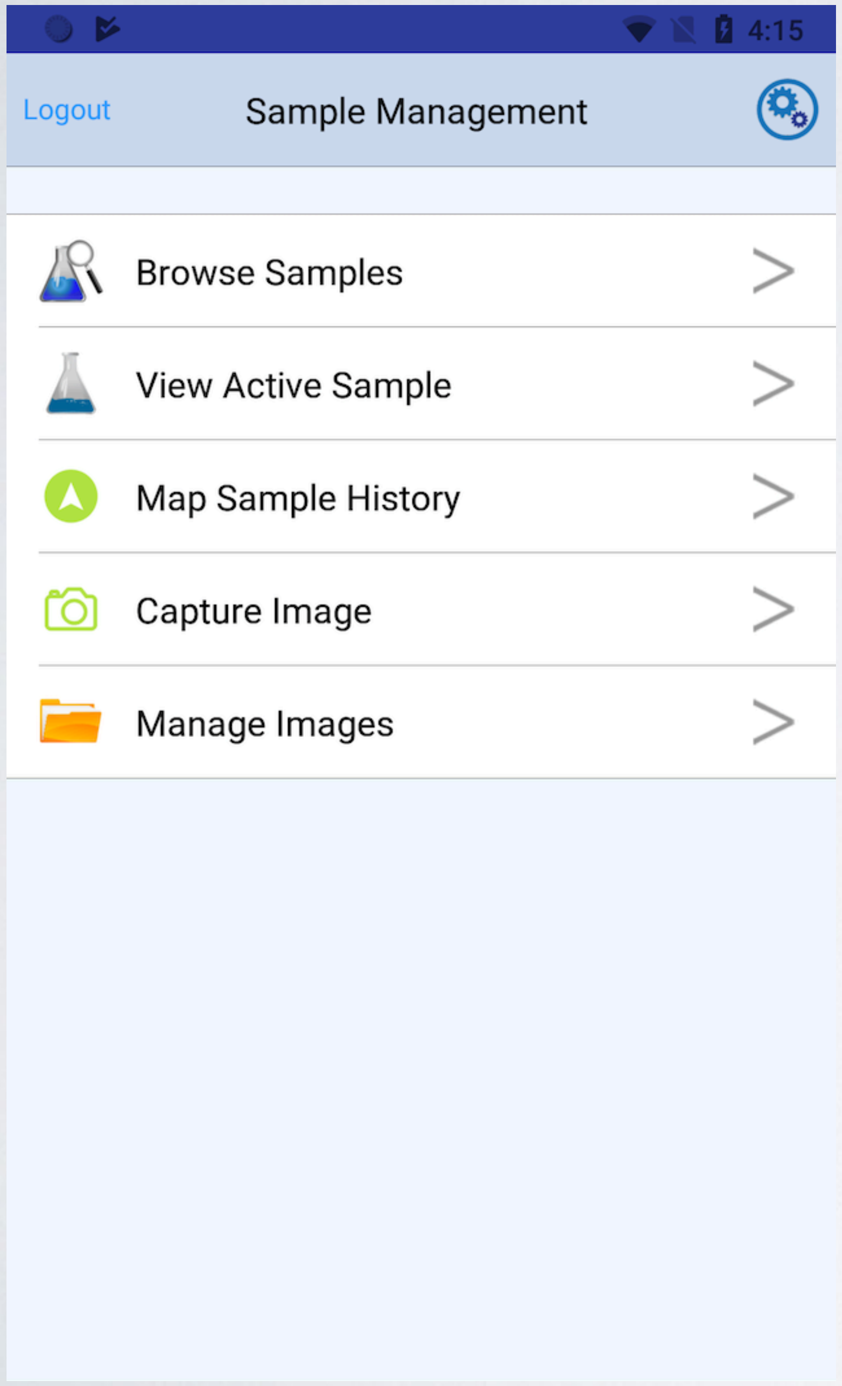
LABWARE MOBILE

- Business logic is the Smalltalk code from LIMS
- Mobile only UI functions also written in Smalltalk
- Native bridge functions replacing LIMS functions for DB, Filesystem etc.
- Mobile only functions for Camera, GPS, MQTT etc.

LABWARE MOBILE

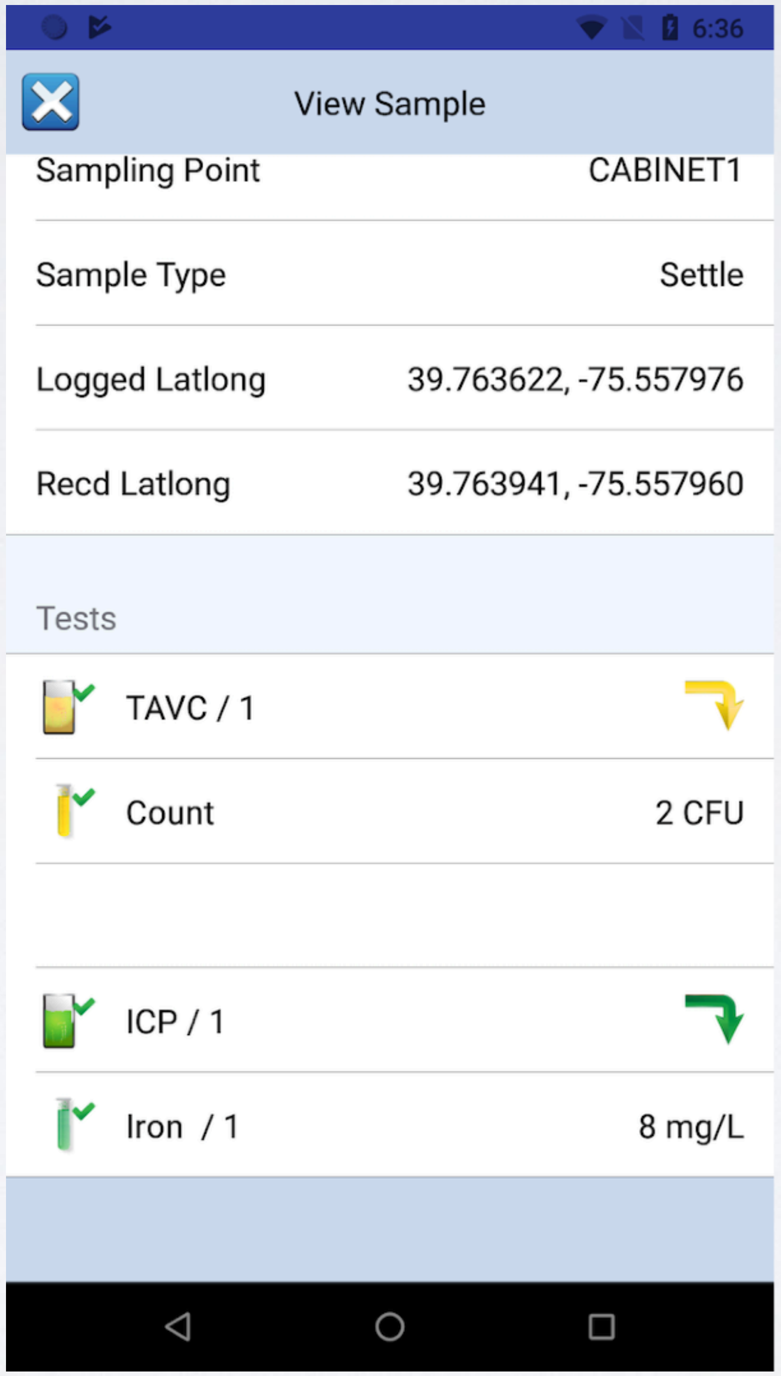
- App provides a toolkit
- Actual app features controlled by user scripts
- Scripts downloaded from server on demand

LABWARE MOBILE



Sample Management

Browse Samples	>
View Active Sample	>
Map Sample History	>
Capture Image	>
Manage Images	>

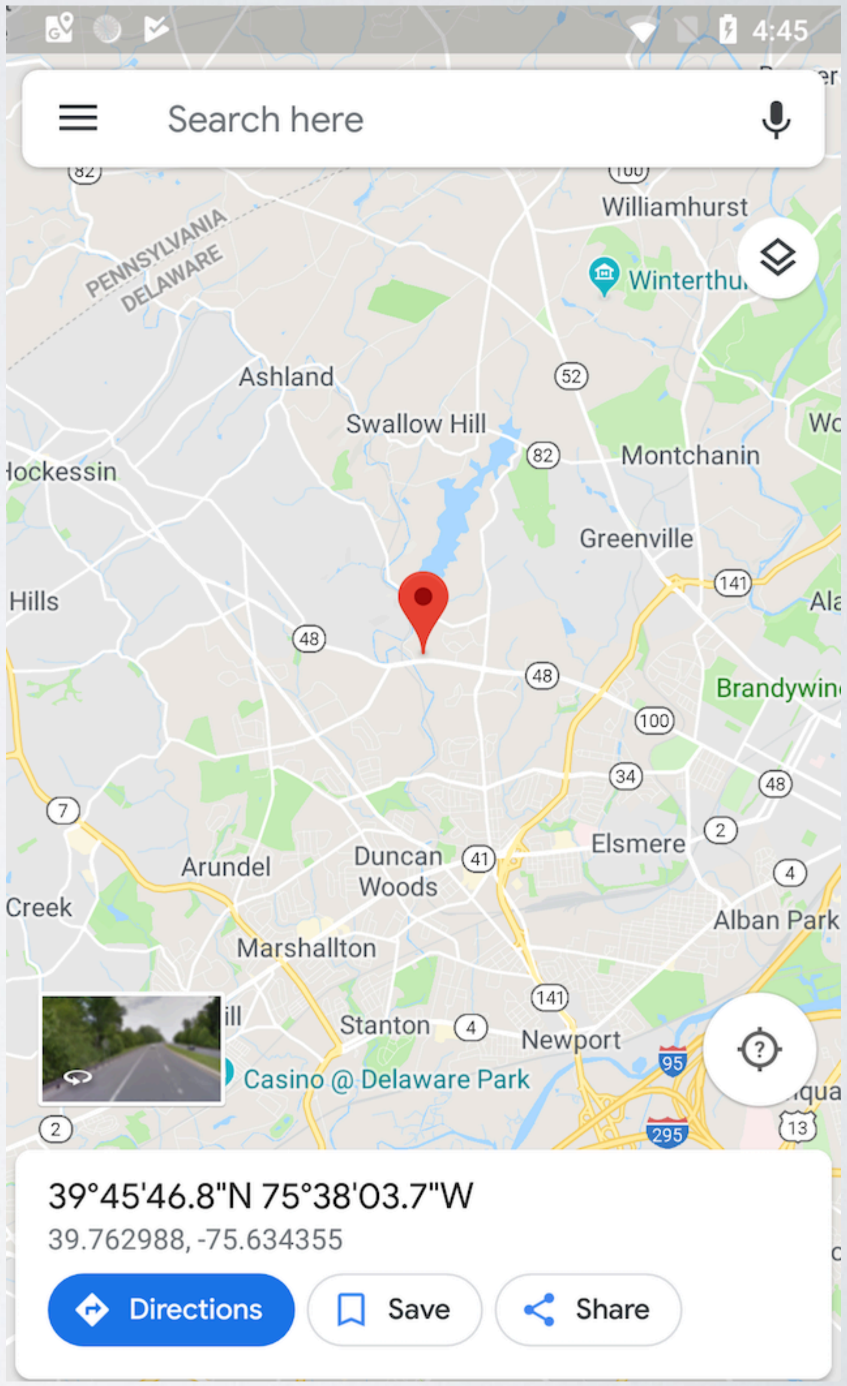


View Sample

Sampling Point	CABINET1
Sample Type	Settle
Logged Latlong	39.763622, -75.557976
Recd Latlong	39.763941, -75.557960

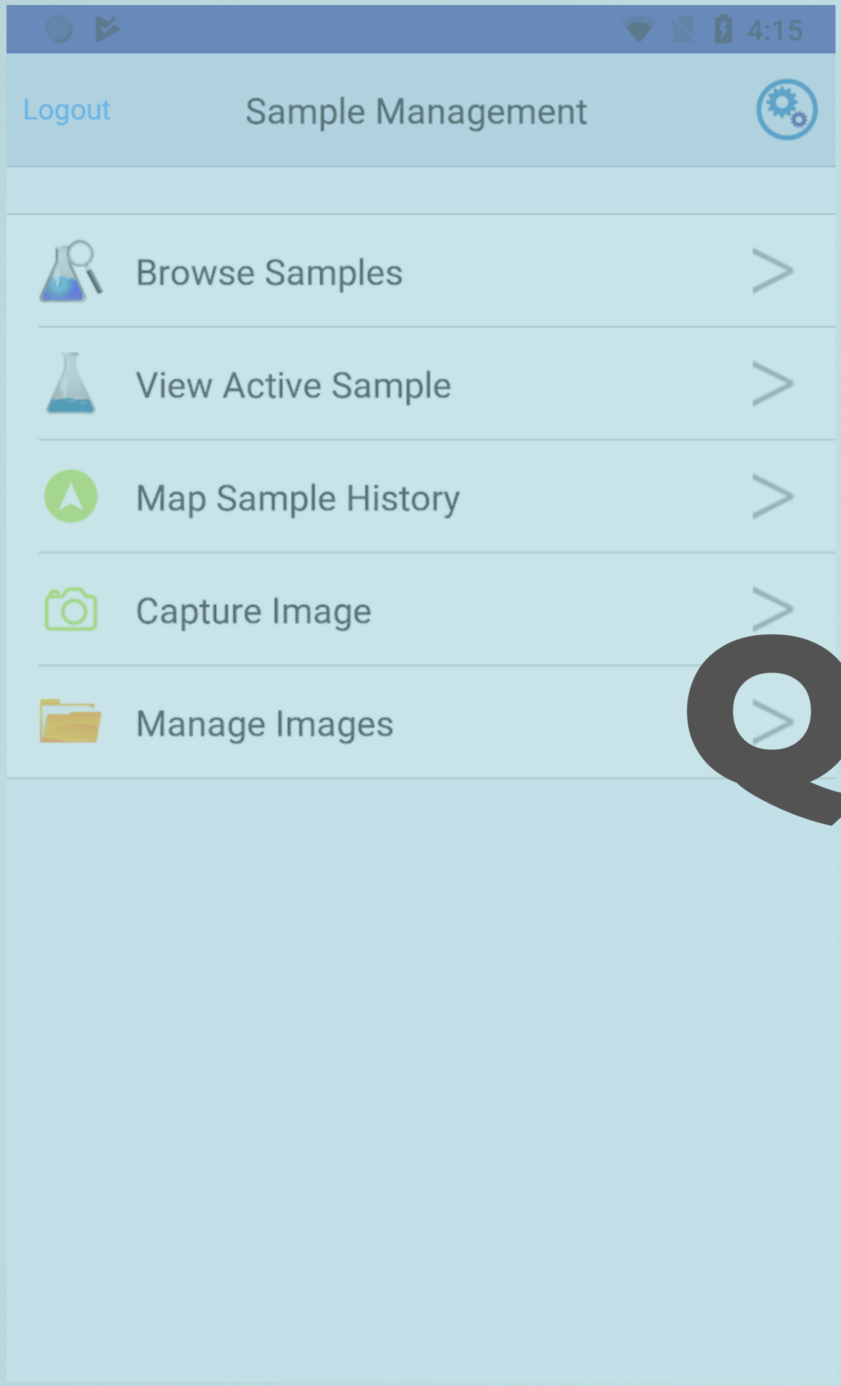
Tests

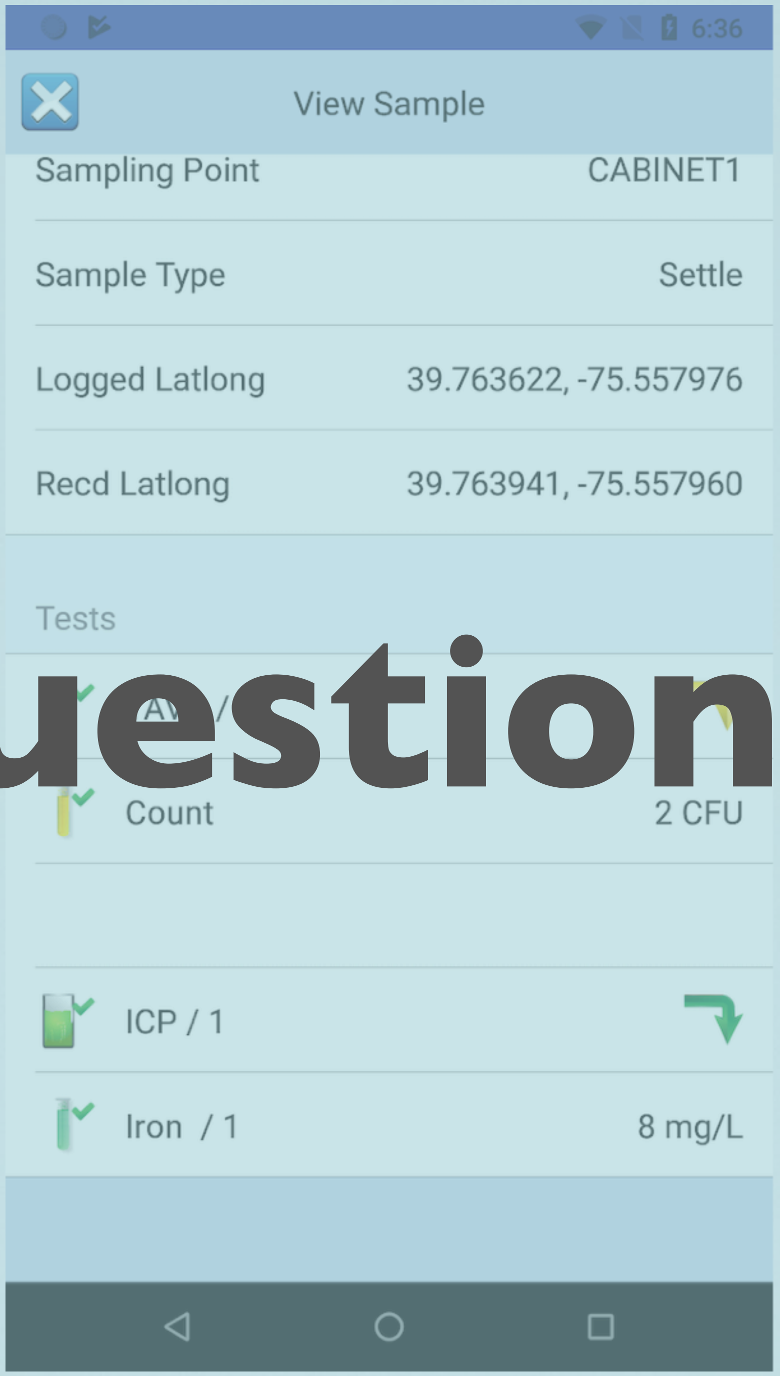
TAVC / 1	
Count	2 CFU
ICP / 1	
Iron / 1	8 mg/L



39°45'46.8"N 75°38'03.7"W
39.762988, -75.634355

Thank you!


4:15
Logout Sample Management
Browse Samples
View Active Sample
Map Sample History
Capture Image
Manage Images


6:36
View Sample

Sampling Point	CABINET1
Sample Type	Settle
Logged Latlong	39.763622, -75.557976
Recd Latlong	39.763941, -75.557960

Tests

Count	2 CFU
ICP / 1	
Iron / 1	8 mg/L

4:45
Search here
39°45'46.8"N 75°38'03.7"W
39.762988, -75.634355

Questions?