

# MOTECH Platform Technical Overview



# The beginnings of MOTECH

## Ghana MOTECH implementation

- Partnered with Ghana Health Service
- Focused on key health needs in most challenging region
- Redesigned existing paper-based systems
- Created two complementary mobile tools
  - Mobile Midwife – actionable IVR/SMS messages to pregnant women and their families
  - Nurses Application – mobile tool to manage and record patient interactions
- Lessons Learned [here](#)

# Turns out we're not alone

Many organizations want to use mobile technology for health

- Drive behavior change & generate demand for services
- Manage patient data
- Improve health worker performance
- Increase patient adherence
- Deliver through the supply chain

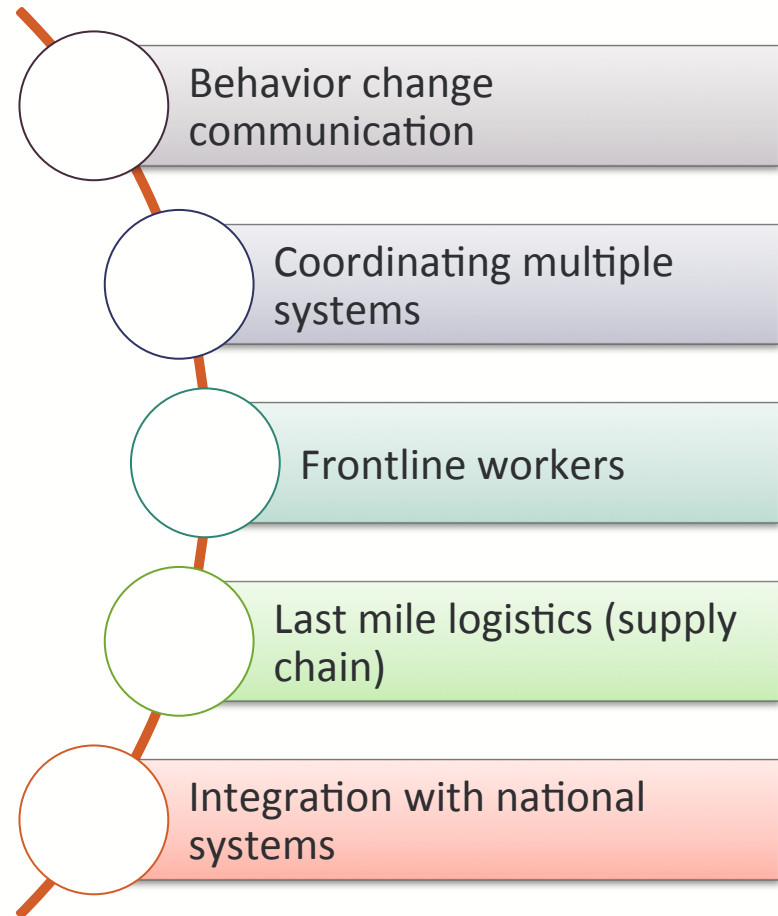
# MOTTECH Platform Focus

- Create general purpose, modular system for mHealth applications
- Support large scale deployments
- Avoid duplication of effort
- Prevent incompatibilities
- Enable implementers to focus on health programs, not software development
- Partner with complementary systems

# What is MOTECH Platform?

## MOTECH Vision

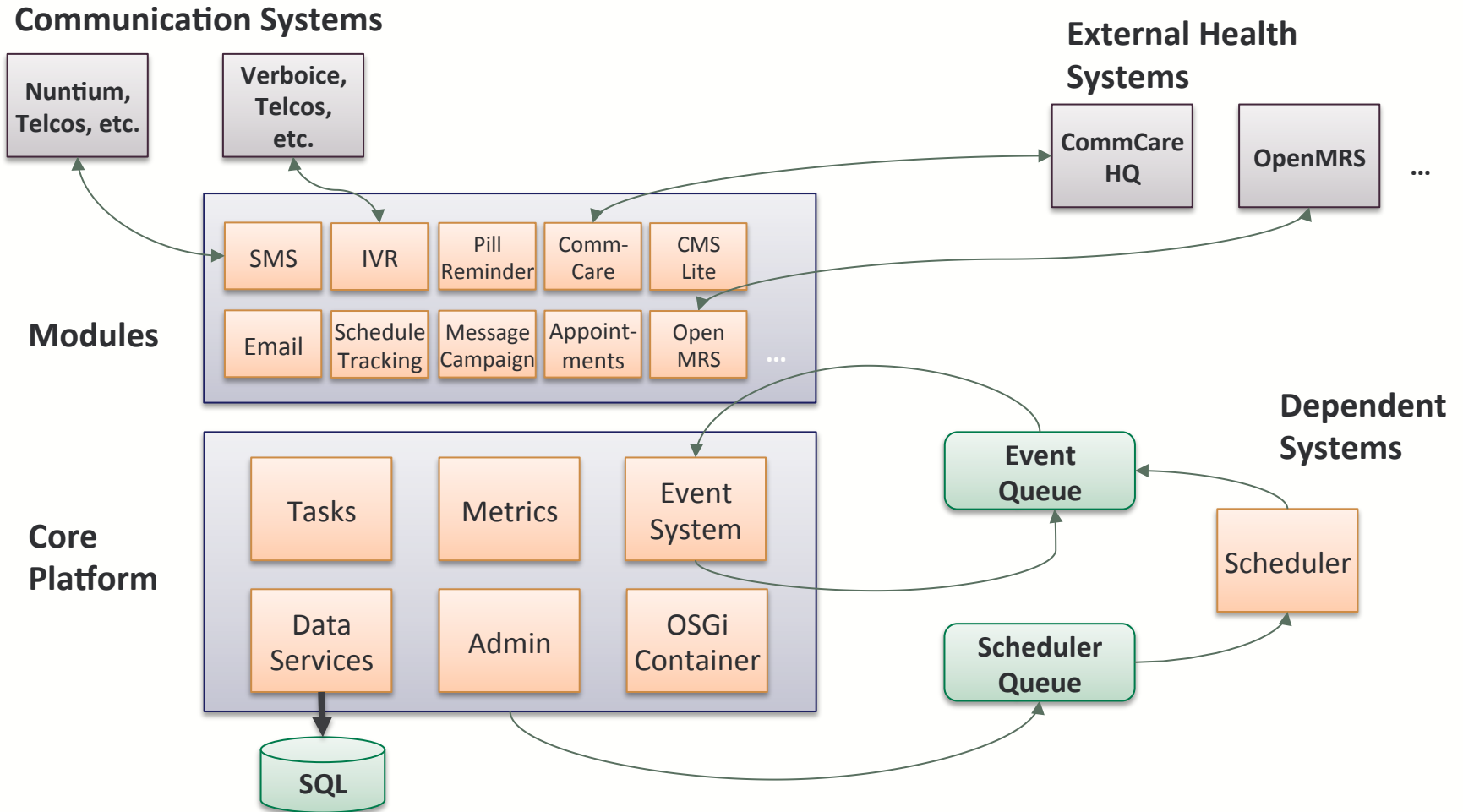
MOTECH combines the integration capabilities of an Enterprise Service Bus (ESB) with a flexible open source application development framework that supports many standard mHealth use cases through its robust, scalable and interoperable core.



# MOTTECH Architecture

- Core Platform
  - Key platform services and API for events, queue, scheduler, data repository
  - UI for administration and security/user management
  - OSGi-based platform, running in Tomcat app server
- Optional Modules
  - Add-in modules for health-domain specific functionality
  - Primary extensibility point for MOTTECH implementers
- OSGi for Lifecycle Management
  - OSGi manages bundle lifecycle (add/remove/start/stop) and allows for exposure of service interfaces

# MOTECH Architecture



# Modules: MOTECH Extensibility

## ■ What is a module?

- OSGi bundle, loaded into server via OSGi host
- Can expose service interface and/or emit events for other modules to consume
- Interact with other modules via their service interfaces or events
- May register servlet controllers to respond to HTTP requests
- May expose data entities via MOTECH Data Services – data browser/editor, REST APIs, record level security, field level auditing
- May expose “triggers” and “actions” to be orchestrated by other modules via Tasks

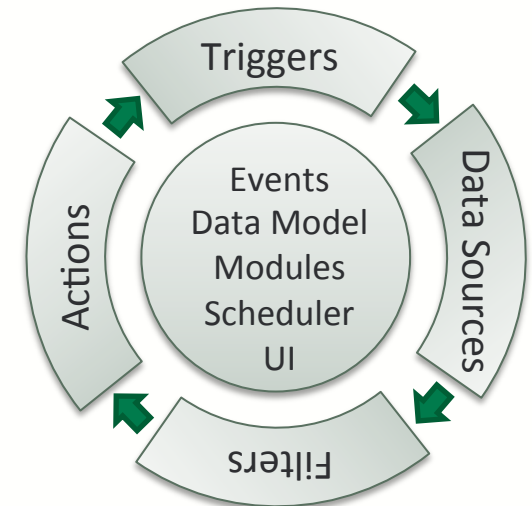
## ■ Why write a module?

- Develop application-specific UI, business logic, data models
- Develop generic reusable functionality to share with MOTECH community



# Core Platform: Tasks System

- Wire up MOTECH modules together without code via a “Task”
  - **Trigger** = event raised by ModuleA (or Scheduler)
  - **Action** = action executed by ModuleB
  - **Filter** = conditional specifying whether or not task should run
- Tasks may look up data from other modules registered as data sources



**Example: When a new patient is created, send her an SMS**

**Trigger** = MRS: new patient

**Action** = SMS: send

**Filter** = patient.phone number != null

# Core Platform: Data Services

- **Problem:** MOTECH implementations devoting piles of code to defining data model (and associated UI, REST APIs, etc.)
- **Solution:** A flexible data modeling system that allows users to define (and share) custom schemas without code
  - UI based entity definition
  - UI for data entry/browsing
  - Generated OSGi services & Java objects for data objects
  - CRUD events
  - Generated REST APIs
  - Auditing & revision tracking
  - All MOTECH modules ported to new data layer



# Modules: Existing

## On Roadmap:

- mTraining
- Batch
- PubSub
- DHIS2
- ETL (Reporting)
- OpenHIE

## Communications

SMS

IVR

Email

Alerts

CMS Lite

Decision Tree

CommCare

Mobile Forms

## Campaigns

Pill Reminder

Schedule Tracking

Message Campaign

## Medical Records

MRS

OpenMRS

Appointments

# Platform Roadmap\*

## Q2 2014

- Complete MOTECH Data Services
- Batch & PubSub Modules
- MOTECH 1.0 Documentation

## Q3 2014

- Semantic Versioning
- UI Normalization
- ETL for integration w/3<sup>rd</sup> party BI

## Q4 2014

- DHIS2 Integration
- OpenHIE Integration
- Upgrade

## 2015

- One-Click Install
- Widget-Based Dashboard
- Integrate with Additional Systems
- MOTECH Containers
- Lightweight Analytics for Non-Tech Users

*\*subject to change, please contact Grameen Foundation for updates on the roadmap*

# Resources

Dev Mailing List	<a href="mailto:motech-dev@googlegroups.com">motech-dev@googlegroups.com</a>
Project Website	<a href="http://www.motechproject.org">http://www.motechproject.org</a>
Code	Platform: <a href="https://code.google.com/p/motech/">https://code.google.com/p/motech/</a> Modules: <a href="http://github.com/motech">http://github.com/motech</a>
Documentation	<a href="http://readthedocs.org">http://readthedocs.org</a> (coming soon)
Issue Tracker	<a href="https://applab.atlassian.net/browse/MOTECH">https://applab.atlassian.net/browse/MOTECH</a>
Dev Machine Setup Instructions	<a href="https://code.google.com/p/motech/wiki/DeveloperMachineSetup">https://code.google.com/p/motech/wiki/DeveloperMachineSetup</a>
Demo Site	<a href="http://zebra.motechcloud.org">http://zebra.motechcloud.org</a> Username: motech / Password: motech
Hello World Tutorial	<a href="https://code.google.com/p/motech/wiki/MotechHelloWorld">https://code.google.com/p/motech/wiki/MotechHelloWorld</a>