Open Data Kit

Building Information Services for Low Income Regions

Yaw Anokwa, Carl Hartung, Waylon Brunette,
Adam Lerer, Clint Tseng, Gaetano Borriello

http://opendatakit.org

My name is Yaw Anokwa. I'm a Ph.D student in Computer Science from the University of Washington and this talk is about Open Data Kit our free and open source set of tools for building information services in low income regions.

This is work that is done with Carl Hartung, Waylon Brunette, Adam Lerer, Clint Tseng, and our advisor Gaetano Borriello. All of us are from UW except for Adam, who is from MIT.

Our team mostly works on healthcare in sub-Saharan Africa, so maybe I can share the kinds of scenarios that inspired ODK.

Paper provides both the information capture and retrieval

When a patient comes into a clinic, the doctor fills out a paper form about the visit.

The paper form goes into a folder, and when the patient comes back a few months later, the doctor reviews the information in that folder and uses that to make decisions.

So paper provides both the information capture and retrieval.

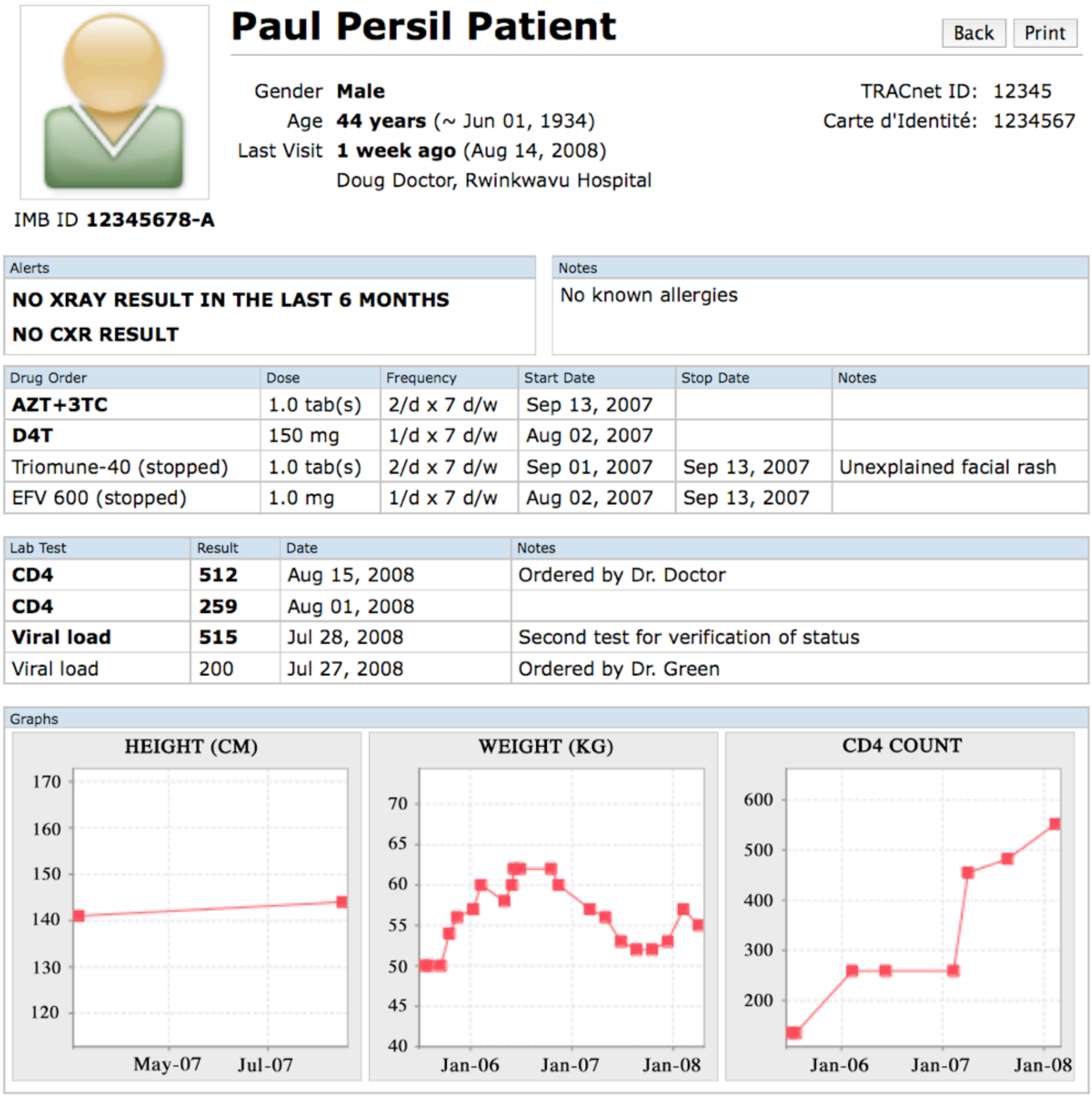
With a paper-based system like this becomes really easy to make clinically relevant mistakes like missing a trend in a record or misread someone's handwriting.

Even if you are really careful, when you get thousand of patients,

Paper-based systems are hard to search or transport

The system becomes hard to search or transport.

This is especially important if you are treating HIV/TB patients who generate a lot of paper records. Ideally, you want to move to an electronic system like this.



Paul Persil Patient

Gender **Male**
Age **44 years** (~ Jun 01, 1934)
Last Visit **1 week ago** (Aug 14, 2008)
Doug Doctor, Rwinkwavu Hospital

TRACnet ID: 12345
Carte d'Identité: 1234567

IMB ID **12345678-A**

Alerts
NO XRAY RESULT IN THE LAST 6 MONTHS
NO CXR RESULT

Notes
No known allergies

Drug Order	Dose	Frequency	Start Date	Stop Date	Notes
AZT+3TC	1.0 tab(s)	2/d x 7 d/w	Sep 13, 2007		
D4T	150 mg	1/d x 7 d/w	Aug 02, 2007		
Triomune-40 (stopped)	1.0 tab(s)	2/d x 7 d/w	Sep 01, 2007	Sep 13, 2007	Unexplained facial rash
EFV 600 (stopped)	1.0 mg	1/d x 7 d/w	Aug 02, 2007	Sep 13, 2007	

Lab Test	Result	Date	Notes
CD4	**512**	Aug 15, 2008	Ordered by Dr. Doctor
CD4	**259**	Aug 01, 2008	
Viral load	**515**	Jul 28, 2008	Second test for verification of status
Viral load	200	Jul 27, 2008	Ordered by Dr. Green

Graphs

Here is a patient summary tool from the OpenMRS electronic medical record system.

You can fill in the data using an electronic device and it gives you the data back in a more useful format complete with clinical alerts and charts.

Beyond the medical domain there are other paper-based systems you might want to consider making more computerized.

If you are a government worker and you need to do a survey in a remote location,

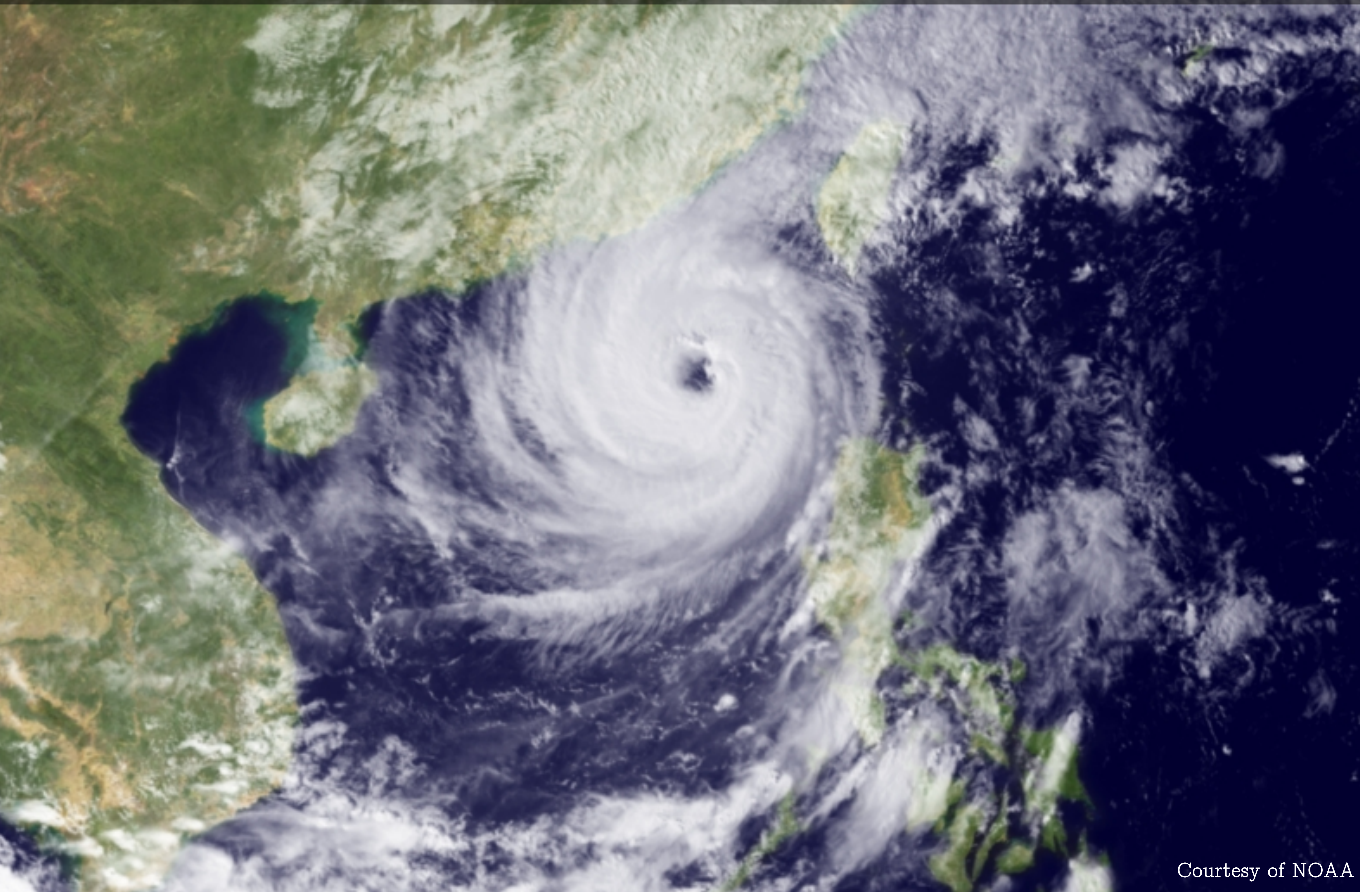
Lag between data collection and actionable information

There is a substantial lag between data collection and actionable information.

That is, if you are in remote desert, it’s going to take you some time to get all those paper forms back to a city so you can get it double entered.

There are also more complex examples of where paper just doesn’t work.

Data collection could provide much richer information

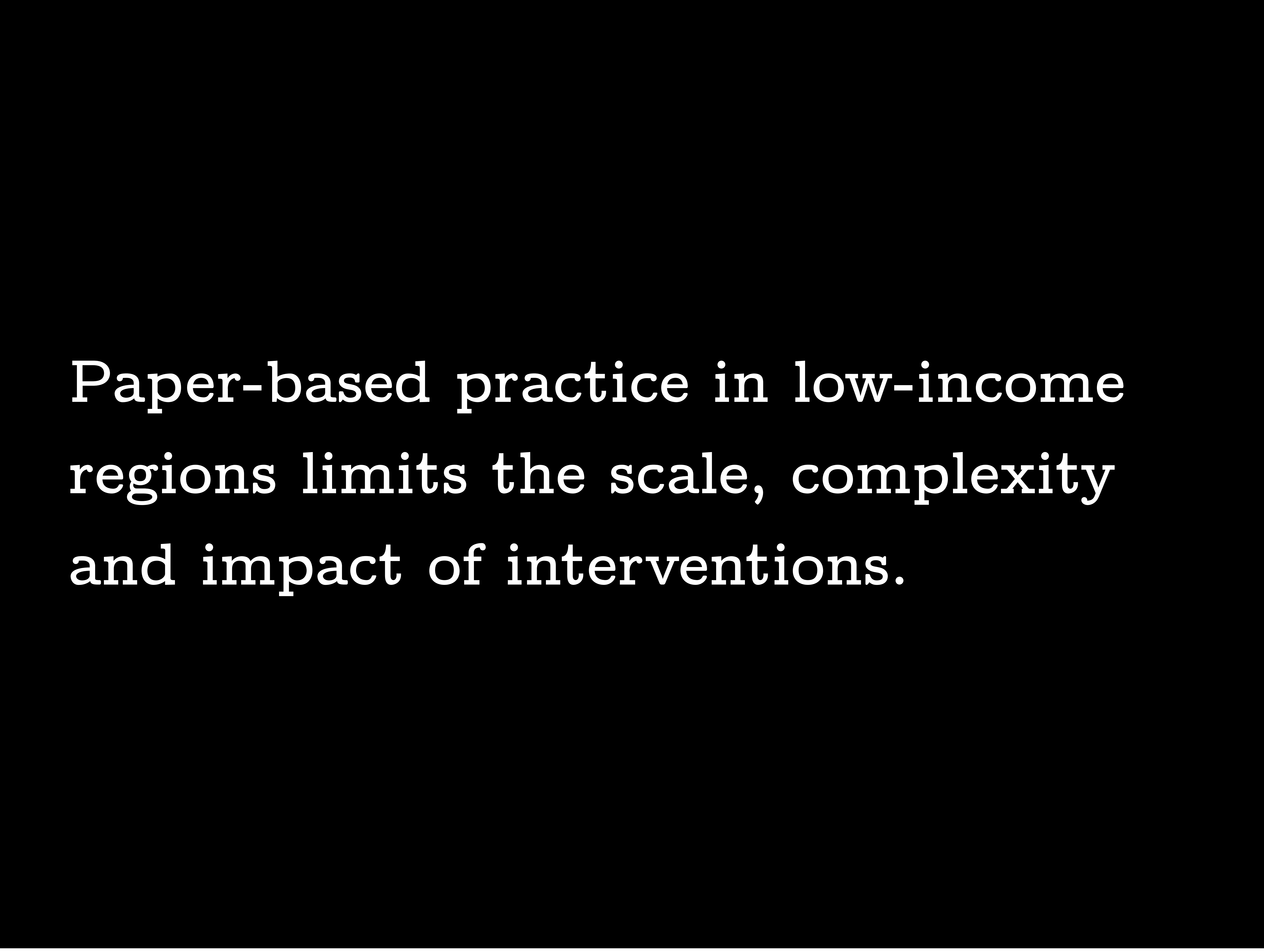


Courtesy of NOAA

If there is a hurricane, and infrastructure has been destroyed, you’d like your data collection to provide much richer information.

For example, you might want to task people to go collect images and GPS locations of damaged buildings.

Basically our claim is



Paper-based practice in low-income regions limits the scale, complexity and impact of interventions.

Paper-based practice in low-income regions limits the scale, complexity and thus the impact of the intervention.

This idea isn’t novel. In fact, Tapan Parikh did a lot of the seminal work in this space.

And with the growth of cell phone usage all over the world, there have been lots of projects where people use phones and computers to replace paper.

We looked at a lot of those tools before building ODK, so let me give you three high level reasons some of those tools are lacking.

Important features are lost when using basic technology

A!
a vitamin
na uwezo wa
kujenga kinga ya mwili dhidi
njwa;
kuimarisha ukuaji wa mimba
ga za majani kwa mafuta ili
ha upatikanaji wa Vitamini A

Staying with the medical domain, here is a picture of a nurse using a PDA to do some decision support.

A lot of projects use PDAs and basic phones. Much of this is cost driven because everyone wants the cheapest device possible...

But. There is a tradeoff between the cost of device and functionality, usability and sustainability of your project and we argue that important features are lost when using basic technology.

For example, on basic phones, you can’t easily take high resolution picture or store a complex medical record system.

Training is also sometimes an issue. We worked with a group who was spending hours just training people on how to capture a location on an external GPS and enter it into a PDA. All that training can be expensive.

We should be able to make that process much easier!

Domain-specific tools are inflexible and keep data siloed

flickr: RaGardner4

The second issue is many of the existing systems are domain-specific and keep the data siloed.

Why can’t we build a modular system that works well for healthcare and crisis mapping and surveying?

Why can’t we make small pieces so each organization can pick just what they need to connect to their existing systems?

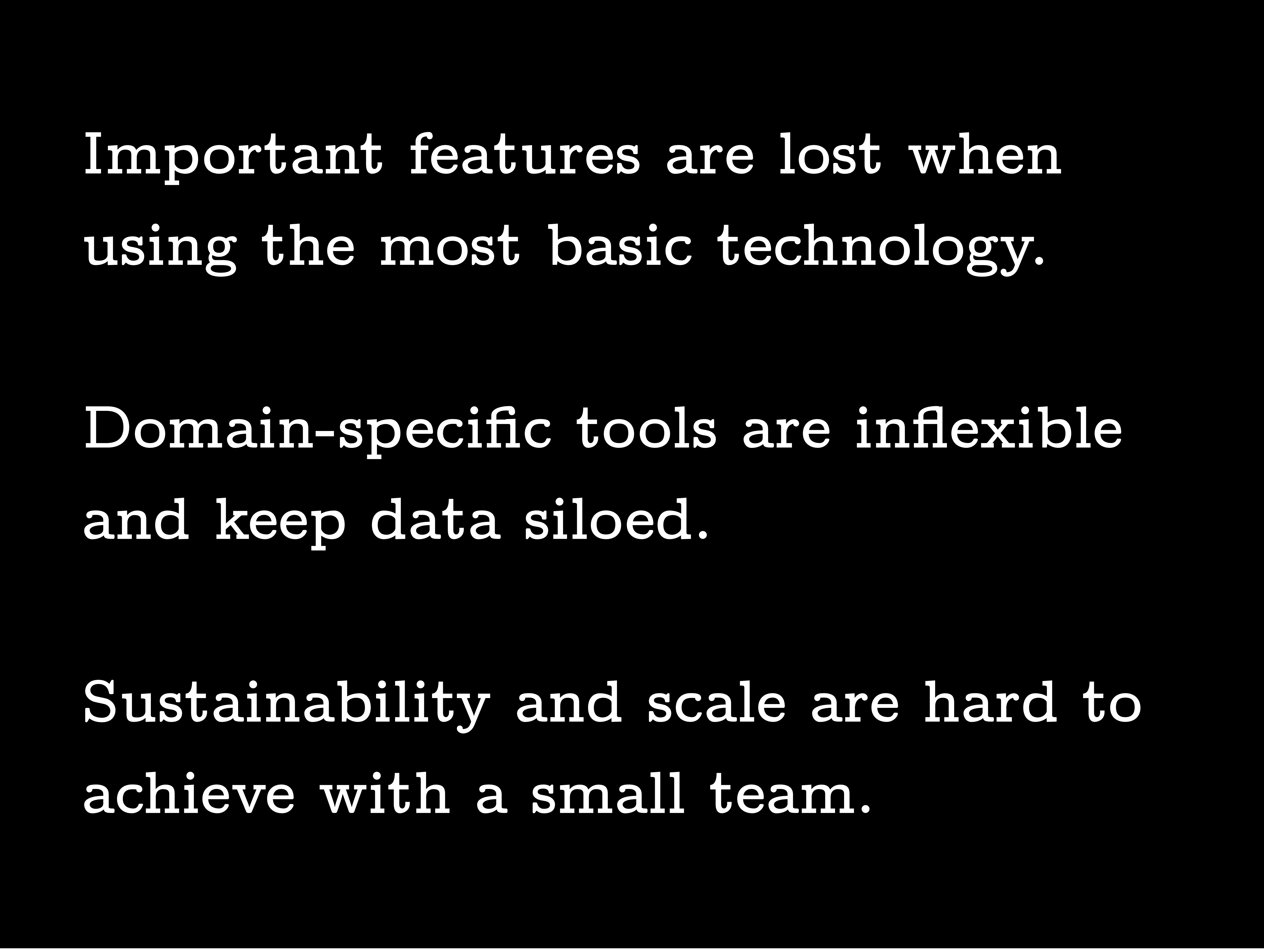
Sustainability is hard to achieve with a small team

And finally, there is the issue of sustainability and scale. Growing a project is hard with small academic team.

Instead of a few people working on ODK, we wanted to build a community of developers and implementers and users and enable that community to grow the project as needed.

This way when we finished the research, the project would continue.

So to recap some of the problems are...


Important features are lost when using the most basic technology.

Domain-specific tools are inflexible and keep data siloed.

Sustainability and scale are hard to achieve with a small team.

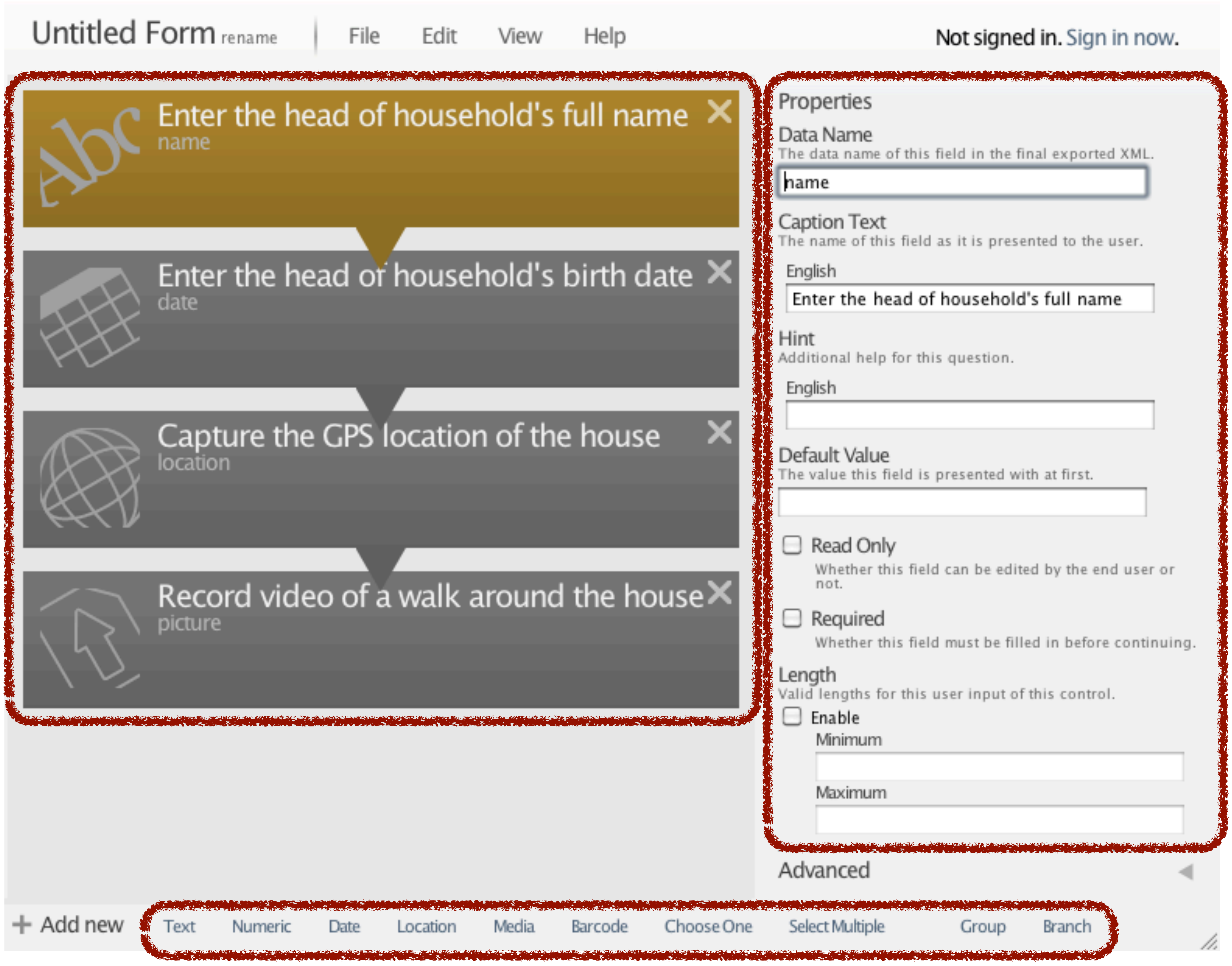
Important features are lost when using the most basic technology.

Domain-specific tools are inflexible and keep data siloed.

Sustainability and scale are hard to achieve with a small team.

So with that, let’s take a look at three of the popular ODK tools.

ODK Build: Drag and drop prompts for form creation



Enter the head of household's full name
name

Enter the head of household's birth date
date

Capture the GPS location of the house
location

Record video of a walk around the house
picture

Properties

Data Name
The data name of this field in the final exported XML.
name

Caption Text
The name of this field as it is presented to the user.
English
Enter the head of household's full name

Hint
Additional help for this question.
English

Default Value
The value this field is presented with at first.

□ Read Only
Whether this field can be edited by the end user or not.

□ Required
Whether this field must be filled in before continuing.

Length
Valid lengths for this user input of this control.
□ Enable
Minimum
Maximum

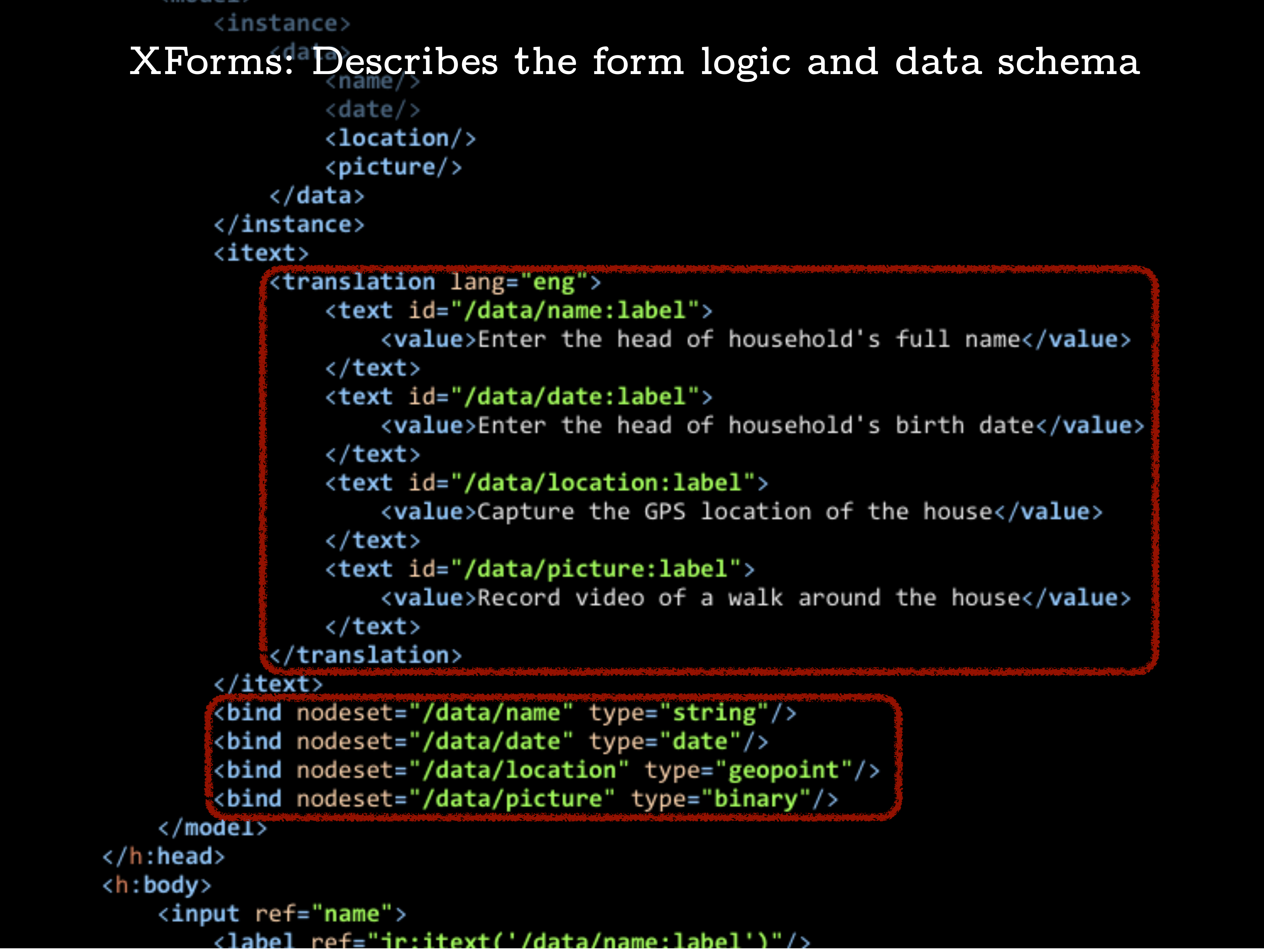
ODK Build is an HTML5 web application where you drag and drop prompts to create forms. It runs in the browser but can also be used offline. It looks like this.

To design a form, you drag and drop each prompt the user will interact with from this button pane to the canvas. Each prompt has a set of properties which users can edit here.

The prompts are pretty powerful. For example, you can have multiple languages for each prompt, make prompts relevant based on previous answers, or make prompts loop some number of times.

When you are done, the Build gives you a file that describes the form logic and the schema of your data.

XForms: Describes the form logic and data schema



```
        <model>
            <instance>
                <data>
                    <name/>
                    <date/>
                    <location/>
                    <picture/>
                </data>
            </instance>
            <itext>
                <translation lang="eng">
                    <text id="/data/name:label">
                        <value>Enter the head of household's full name</value>
                    </text>
                    <text id="/data/date:label">
                        <value>Enter the head of household's birth date</value>
                    </text>
                    <text id="/data/location:label">
                        <value>Capture the GPS location of the house</value>
                    </text>
                    <text id="/data/picture:label">
                        <value>Record video of a walk around the house</value>
                    </text>
                </translation>
            </itext>
            <bind nodeset="/data/name" type="string"/>
            <bind nodeset="/data/date" type="date"/>
            <bind nodeset="/data/location" type="geopoint"/>
            <bind nodeset="/data/picture" type="binary"/>
        </model>
    </h:head>
    <h:body>
        <input ref="name">
            <label ref="jr:itext('/data/name:label')"/>
```

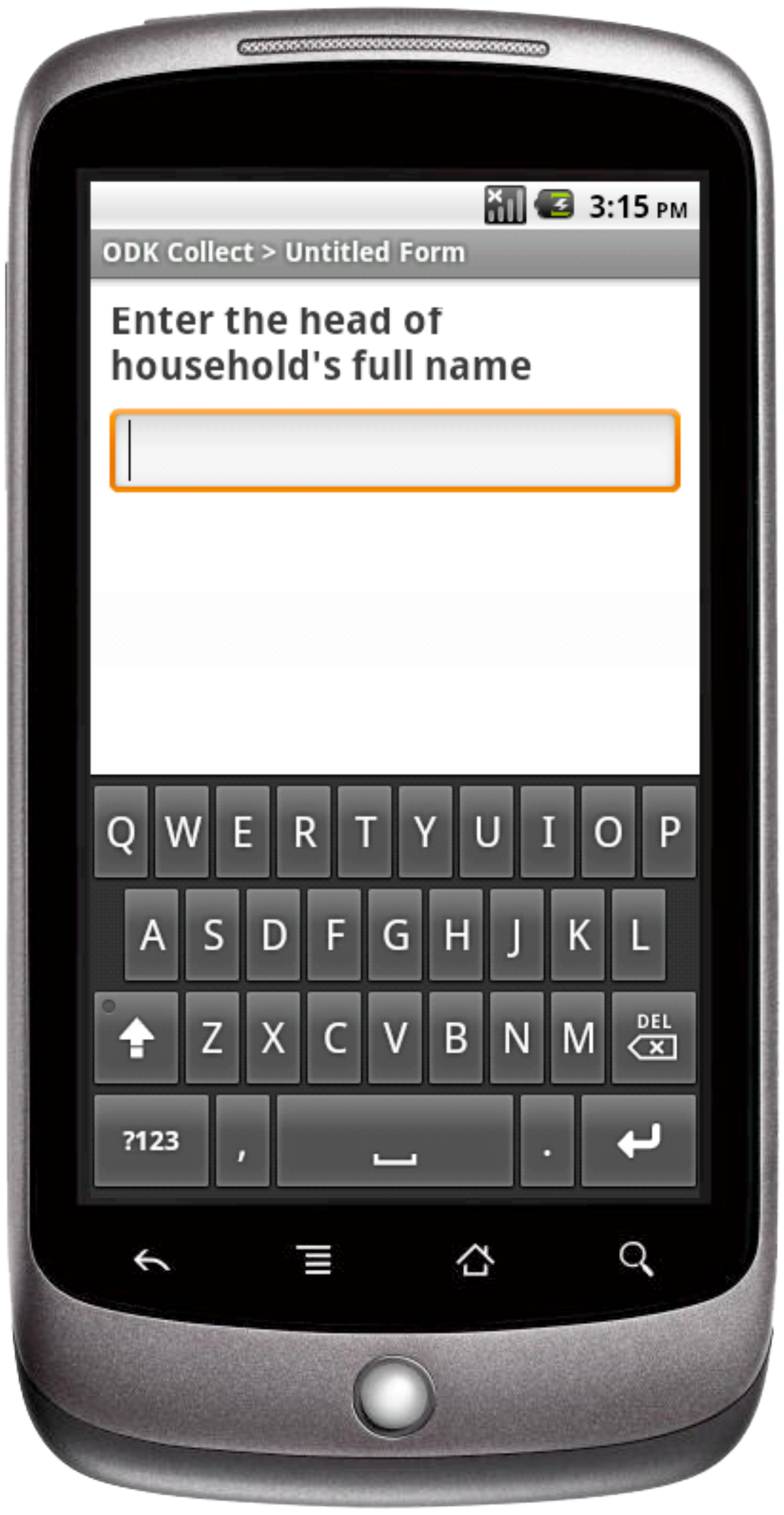
That file is an XForm. W3C XForms are a well-defined open standard for forms and we use it to describe the form logic and data schema.

If you can read some of this text, you'll see some of the elements of the form I showed you earlier, like the english translation for the text.

You see also data types -- string, date, geopoint and binary.

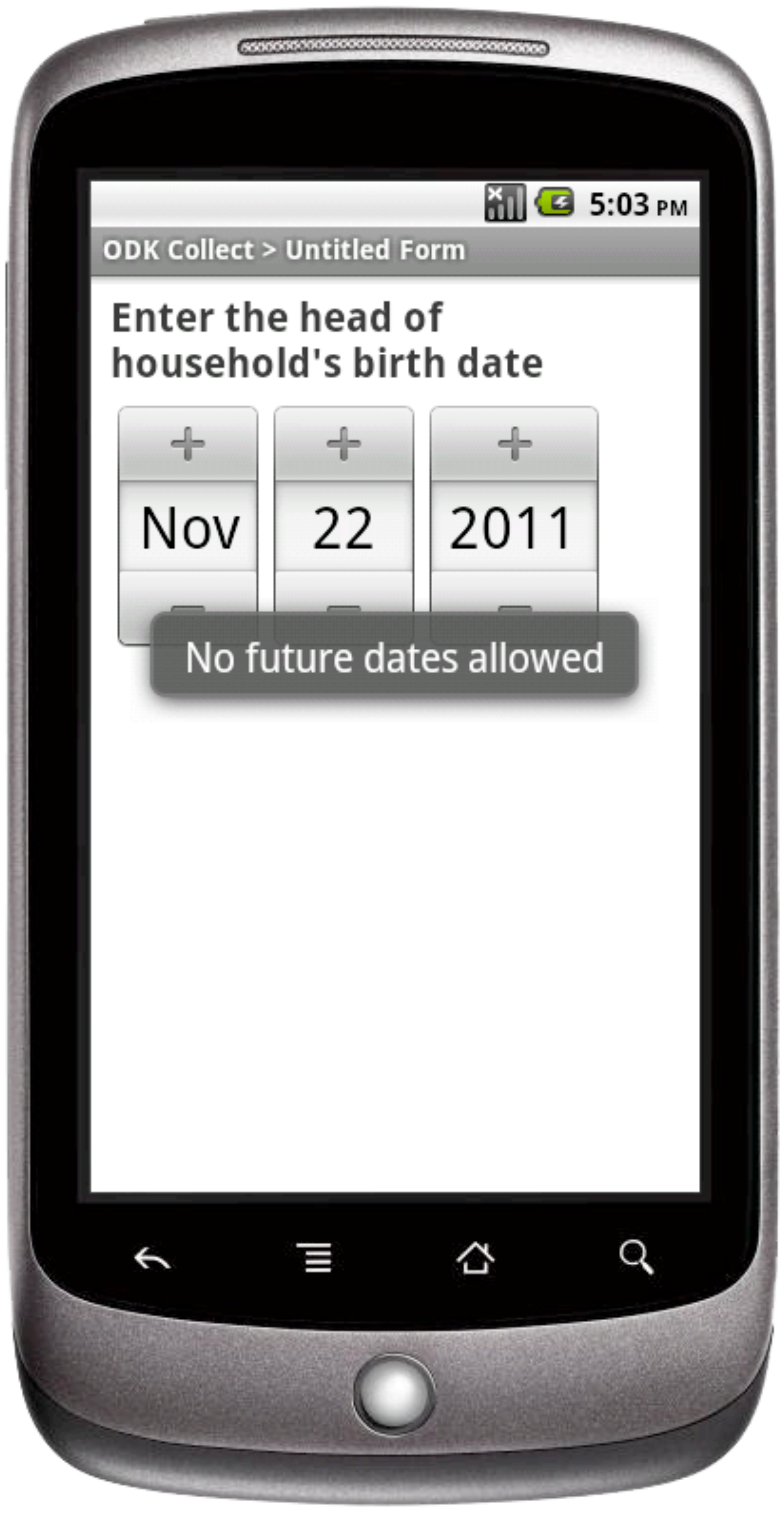
Once you have a form, you can give it to a device you want to show it -- like our mobile client, ODK Collect.

Collect: Display prompts for data collection and delivery



ODK Collect > Untitled Form

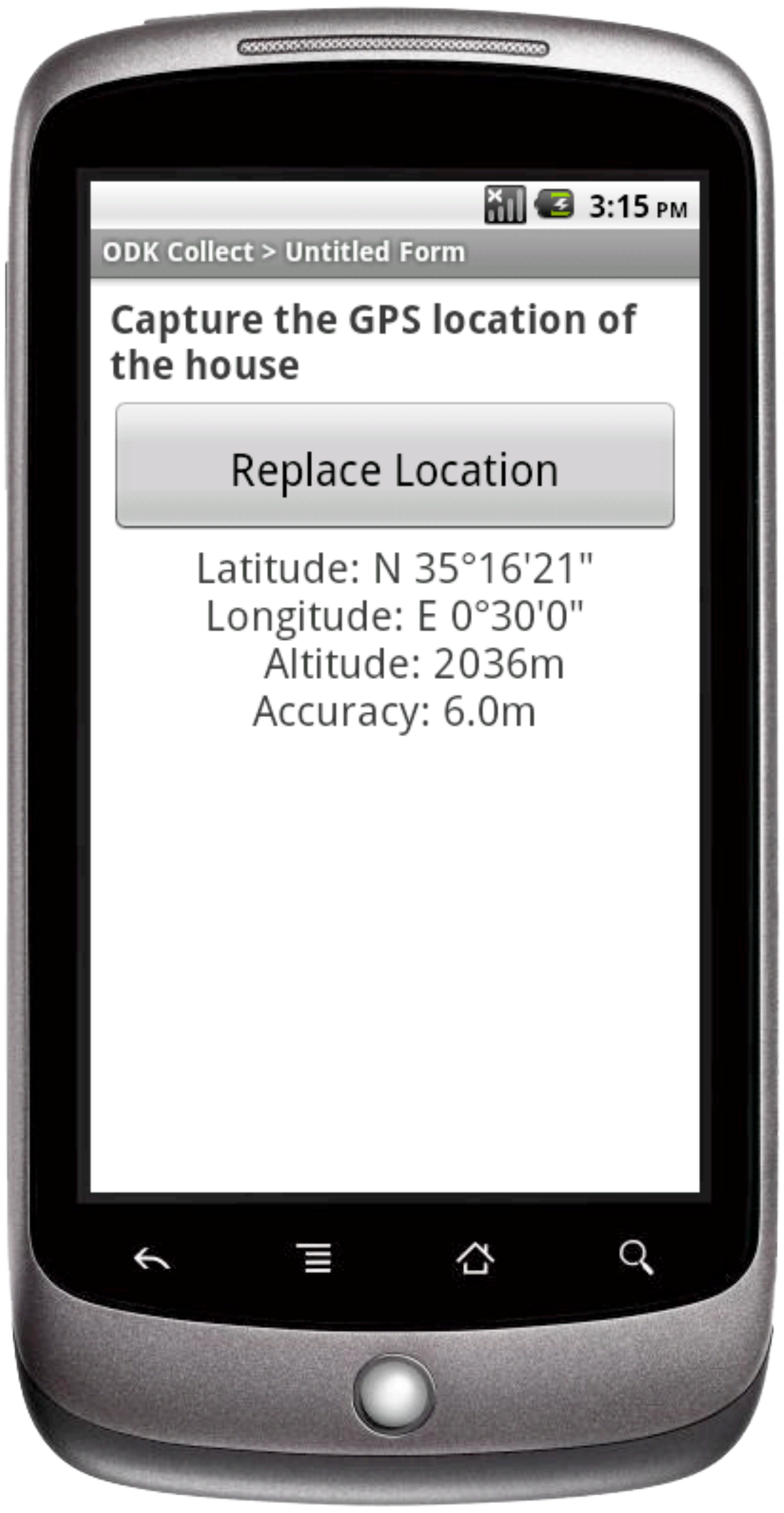
Enter the head of household's full name



ODK Collect > Untitled Form

Enter the head of household's birth date

No future dates allowed



ODK Collect > Untitled Form

Capture the GPS location of the house

Latitude: N 35°16'21"
Longitude: E 0°30'0"
Altitude: 2036m
Accuracy: 6.0m

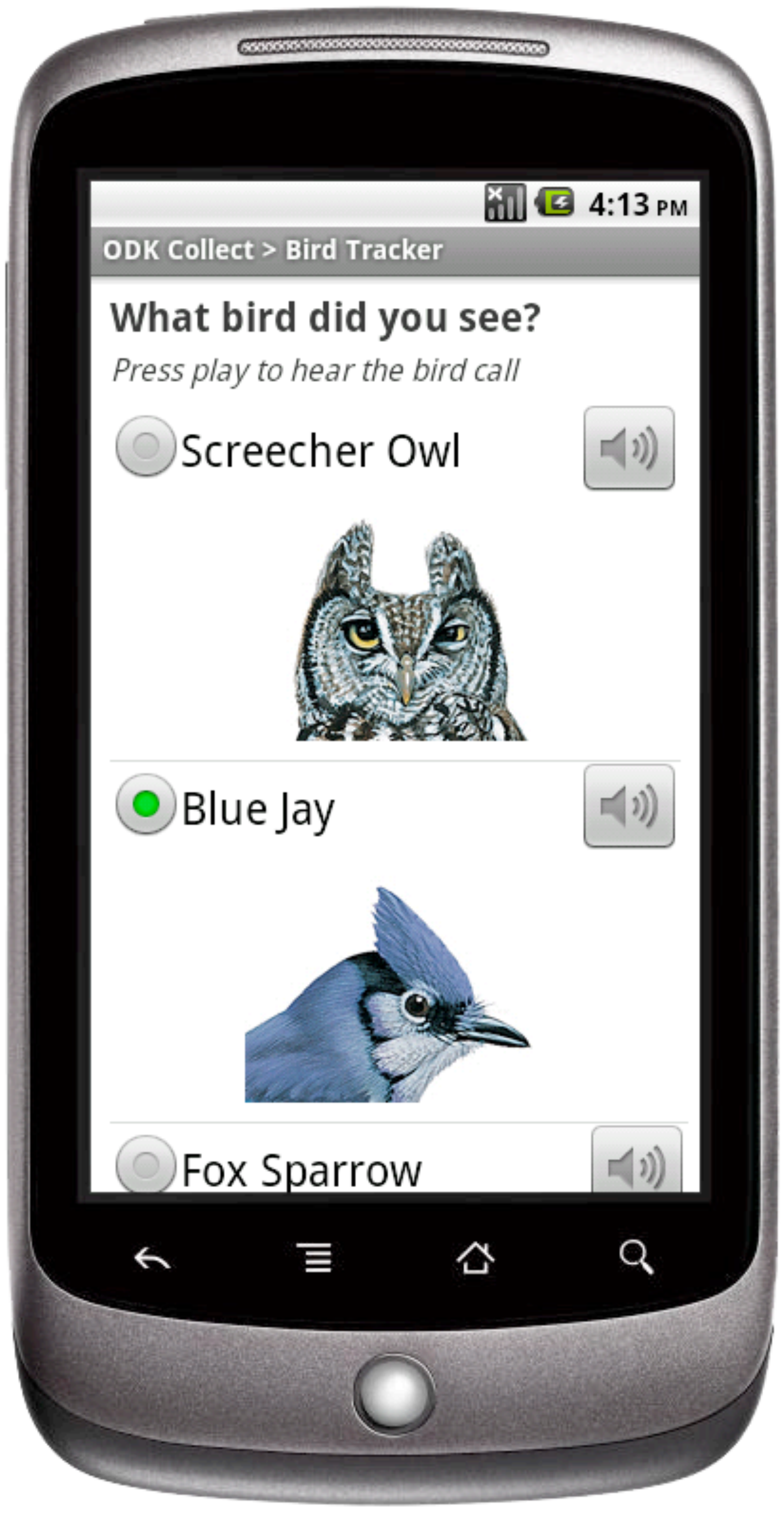
ODK Collect is an client that takes the XForm and can display prompts to collect and deliver different types of data.

Collect runs on the Android operating system so you can use it on phones, tablets and netbooks. It runs great offline and looks like this.

You can collect text, numbers, and dates. You can put constraints on all prompts. So for example, birth dates can’t be in the future.

You can also capture GPS location using one click.

Collect: Render prompts for data collection and delivery


4:13 PM

ODK Collect > Bird Tracker

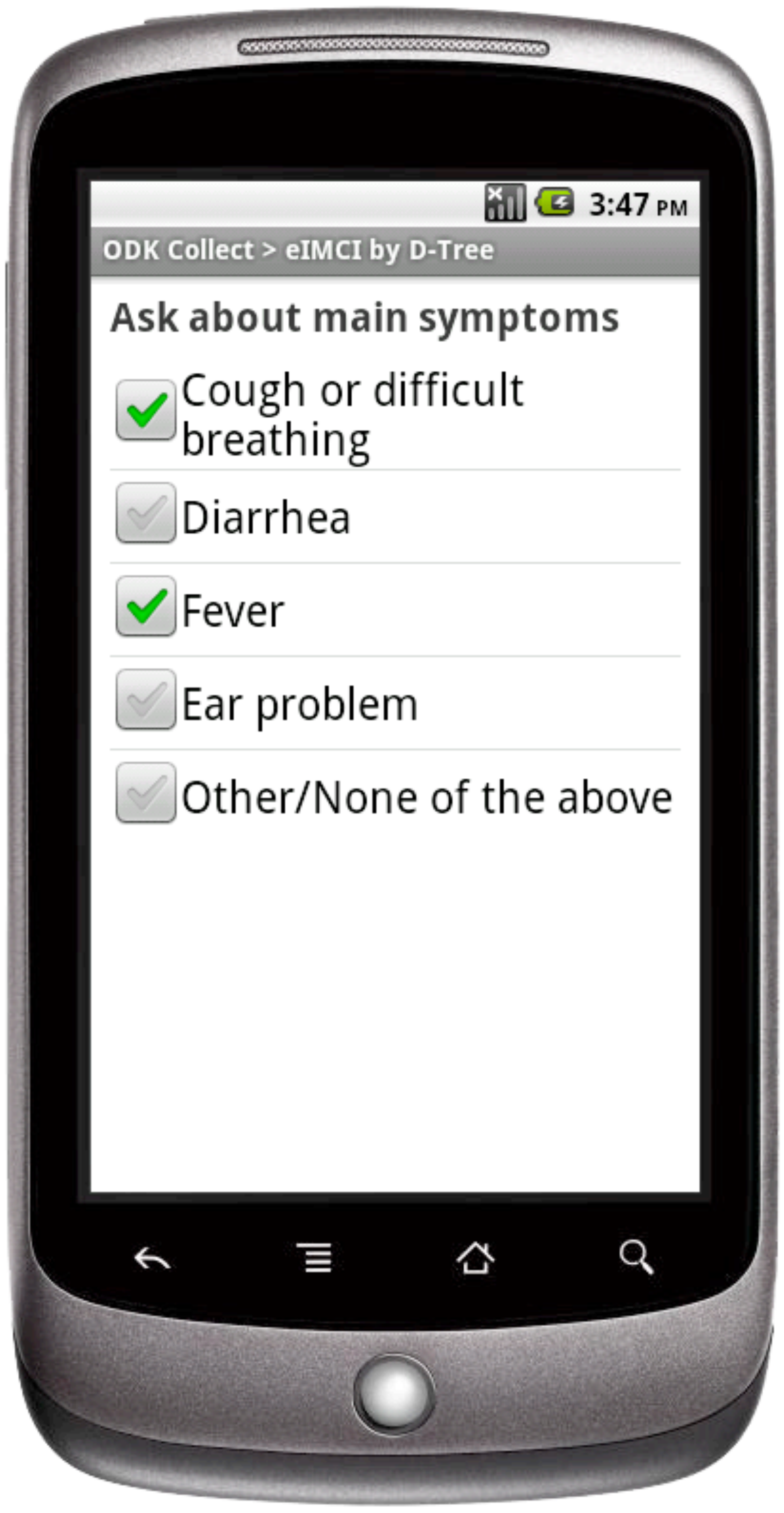
What bird did you see?

Press play to hear the bird call

○ Screecher Owl

● Blue Jay

○ Fox Sparrow


3:47 PM

ODK Collect > eIMCI by D-Tree

Ask about main symptoms

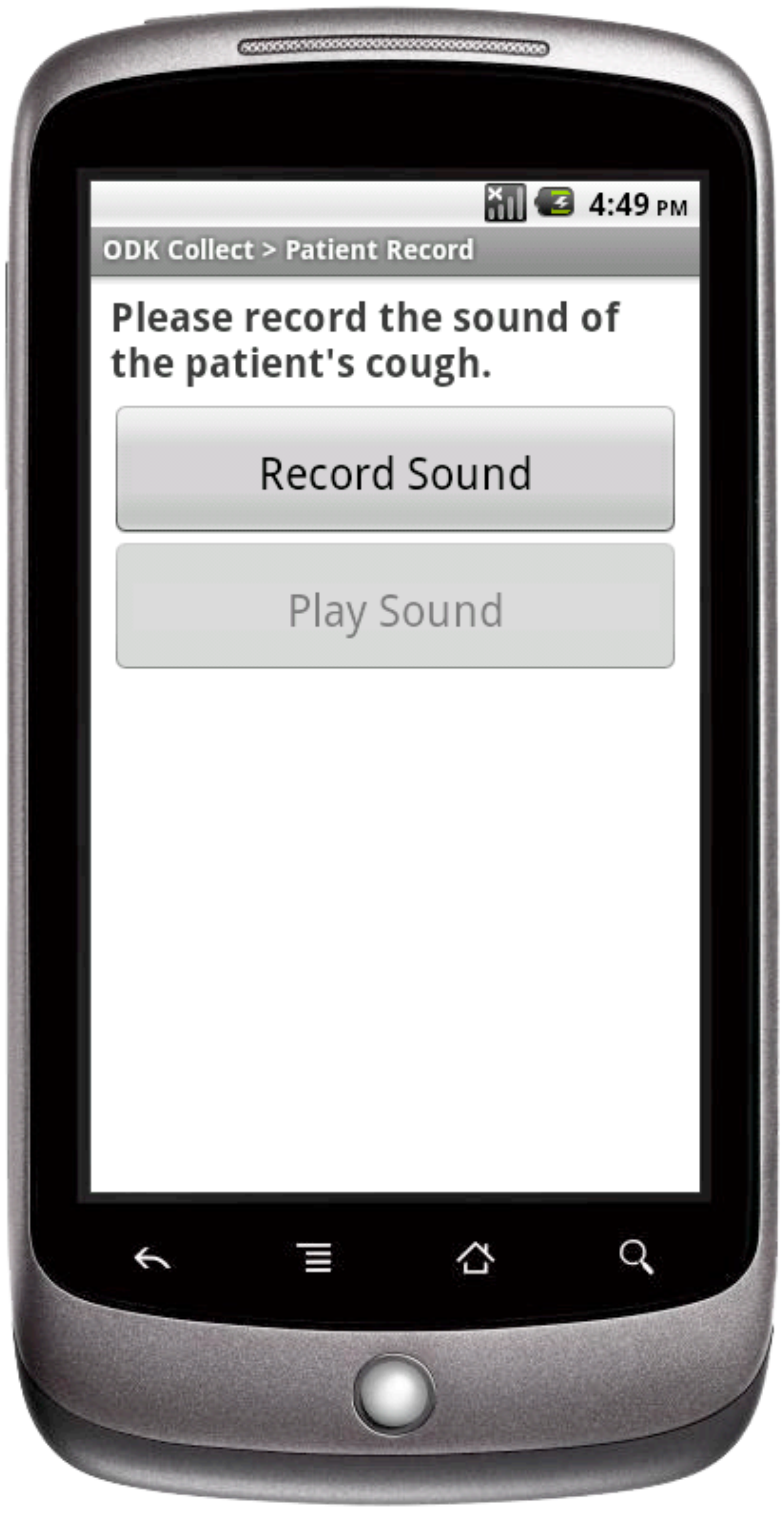
☑ Cough or difficult breathing

□ Diarrhea

☑ Fever

□ Ear problem

□ Other/None of the above


4:49 PM

ODK Collect > Patient Record

Please record the sound of the patient's cough.

The prompts can be multimedia rich. In this example, you can both see the bird and play the sound of each bird before continuing.

The prompts can be translated into any language (say Swahili). You can switch them on the fly (to say English).

The prompts can have logic and are powerful enough to do clinical decision support. This is what we mean by delivering information.

This form is from the WHO’s diagnosis protocol for sick children under five. As you tell the phone the symptoms, it can suggest treatments.

You can combine this logic with the multimedia. So if say the child is coughing or has a fever, ODK Collect can record the sound of a cough or even play a video about how to treat fever.

Collect: Extensible architecture built on Android

One of the things that makes Collect unique is that it has an extensible architecture built around Android.

So for example, when one of our users wanted barcode scanning of patient ids, we didn’t have to write a barcode scanner.

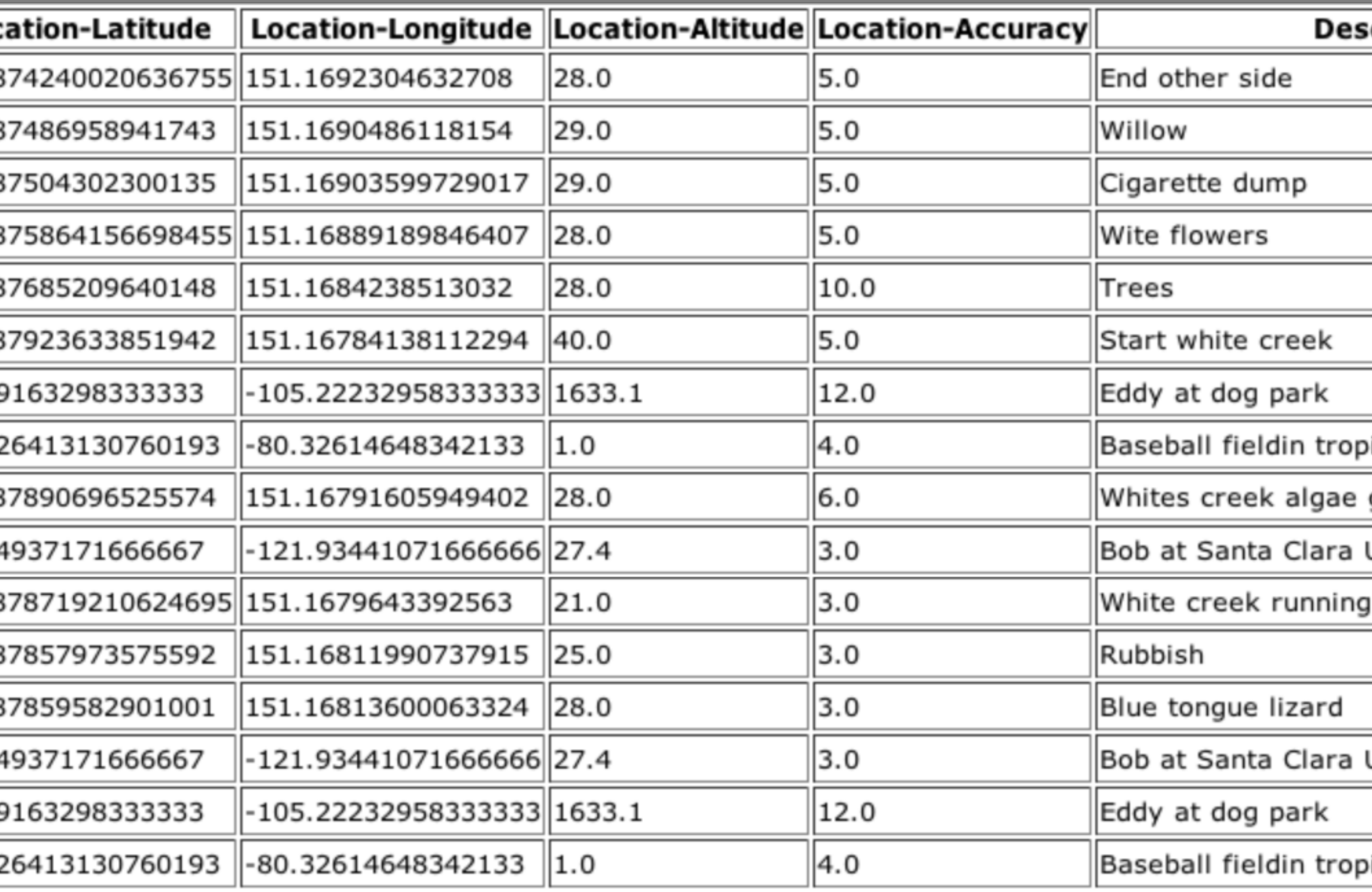
You tell the Android OS you need to scan a barcode and if there is an application that can do that, it starts up, scans the barcode and gives you back the response.

This leveraging of existing apps is impossible on other platforms and it saves us a lot of work.

Once you are done with the prompts, all the data is saved offline. You can, if you choose to, send it to any server over USB, the cell network or WiFi.

If you don’t have a server, we have one called ODK Aggregate.

Aggregate: Host data and provide extraction interfaces



cation-Latitude	Location-Longitude	Location-Altitude	Location-Accuracy	Des
374240020636755	151.1692304632708	28.0	5.0	End other side
37486958941743	151.1690486118154	29.0	5.0	Willow
37504302300135	151.16903599729017	29.0	5.0	Cigarette dump
375864156698455	151.16889189846407	28.0	5.0	Wite flowers
37685209640148	151.1684238513032	28.0	10.0	Trees
37923633851942	151.16784138112294	40.0	5.0	Start white creek
9163298333333	-105.22232958333333	1633.1	12.0	Eddy at dog park
26413130760193	-80.32614648342133	1.0	4.0	Baseball fieldin trop
37890696525574	151.16791605949402	28.0	6.0	Whites creek algae
4937171666667	-121.93441071666666	27.4	3.0	Bob at Santa Clara
378719210624695	151.1679643392563	21.0	3.0	White creek running
37857973575592	151.16811990737915	25.0	3.0	Rubbish
37859582901001	151.16813600063324	28.0	3.0	Blue tongue lizard
4937171666667	-121.93441071666666	27.4	3.0	Bob at Santa Clara
9163298333333	-105.22232958333333	1633.1	12.0	Eddy at dog park
26413130760193	-80.32614648342133	1.0	4.0	Baseball fieldin trop

Aggregate hosts the submitted data and provides extraction interfaces such as spreadsheets, maps, and queries.

We don’t run one big server, you download an installer, and it configures one just for your organization.

Aggregate: Codebase runs locally and in the cloud

flickr: theplanetdotcom

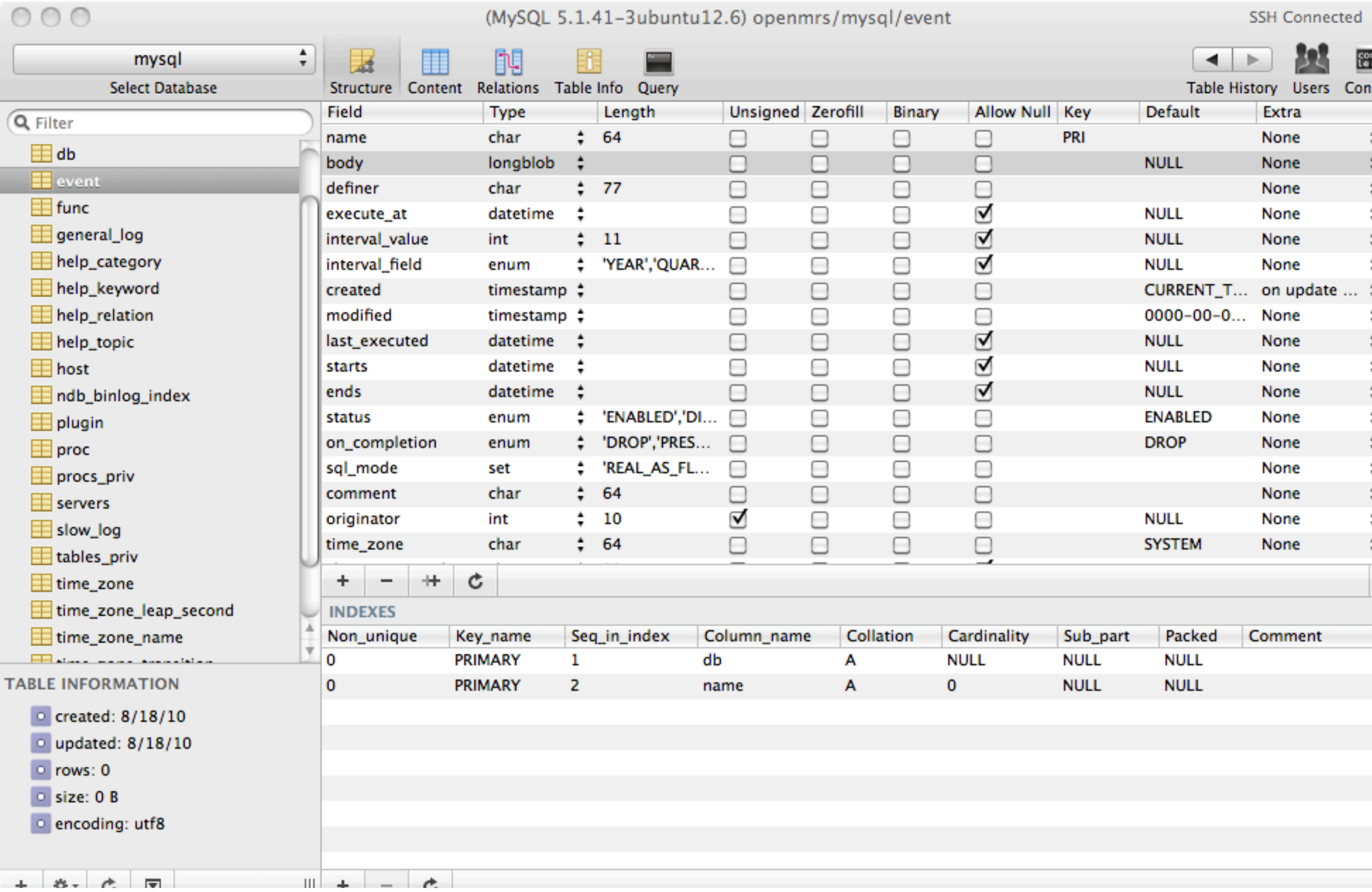
Aggregate is designed to run on local machines as well as run in the cloud.

In fact, the same code base that runs on a local server backed with MySQL will run on Google App Engine or Amazon Web Services.

The benefit of the cloud over the local server is with a double click and you get a fast server that is basically free.

You don't have to worry about viruses, maintenance, scaling, or the other problems of running a server in a low-income region.

Aggregate: Designs database backend using XForm



db
event
func
general_log
help_category
help_keyword
help_relation
help_topic
host
ndb_binlog_index
plugin
proc
procs_priv
servers
slow_log
tables_priv
time_zone
time_zone_leap_second
time_zone_name

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra
name	char	64	☐	☐	☐	☐	PRI		None
body	longblob		☐	☐	☐	☐		NULL	None
definer	char	77	☐	☐	☐	☐			None
execute_at	datetime		☐	☐	☐	☑		NULL	None
interval_value	int	11	☐	☐	☐	☑		NULL	None
interval_field	enum	'YEAR','QUAR...	☐	☐	☐	☑		NULL	None
created	timestamp		☐	☐	☐	☐		CURRENT_T...	on update ...
modified	timestamp		☐	☐	☐	☐		0000-00-0...	None
last_executed	datetime		☐	☐	☐	☑		NULL	None
starts	datetime		☐	☐	☐	☑		NULL	None
ends	datetime		☐	☐	☐	☑		NULL	None
status	enum	'ENABLED','DI...	☐	☐	☐	☐		ENABLED	None
on_completion	enum	'DROP','PRES...	☐	☐	☐	☐		DROP	None
sql_mode	set	'REAL_AS_FL...	☐	☐	☐	☐			None
comment	char	64	☐	☐	☐	☐			None
originator	int	10	☑	☐	☐	☐		NULL	None
time_zone	char	64	☐	☐	☐	☐		SYSTEM	None

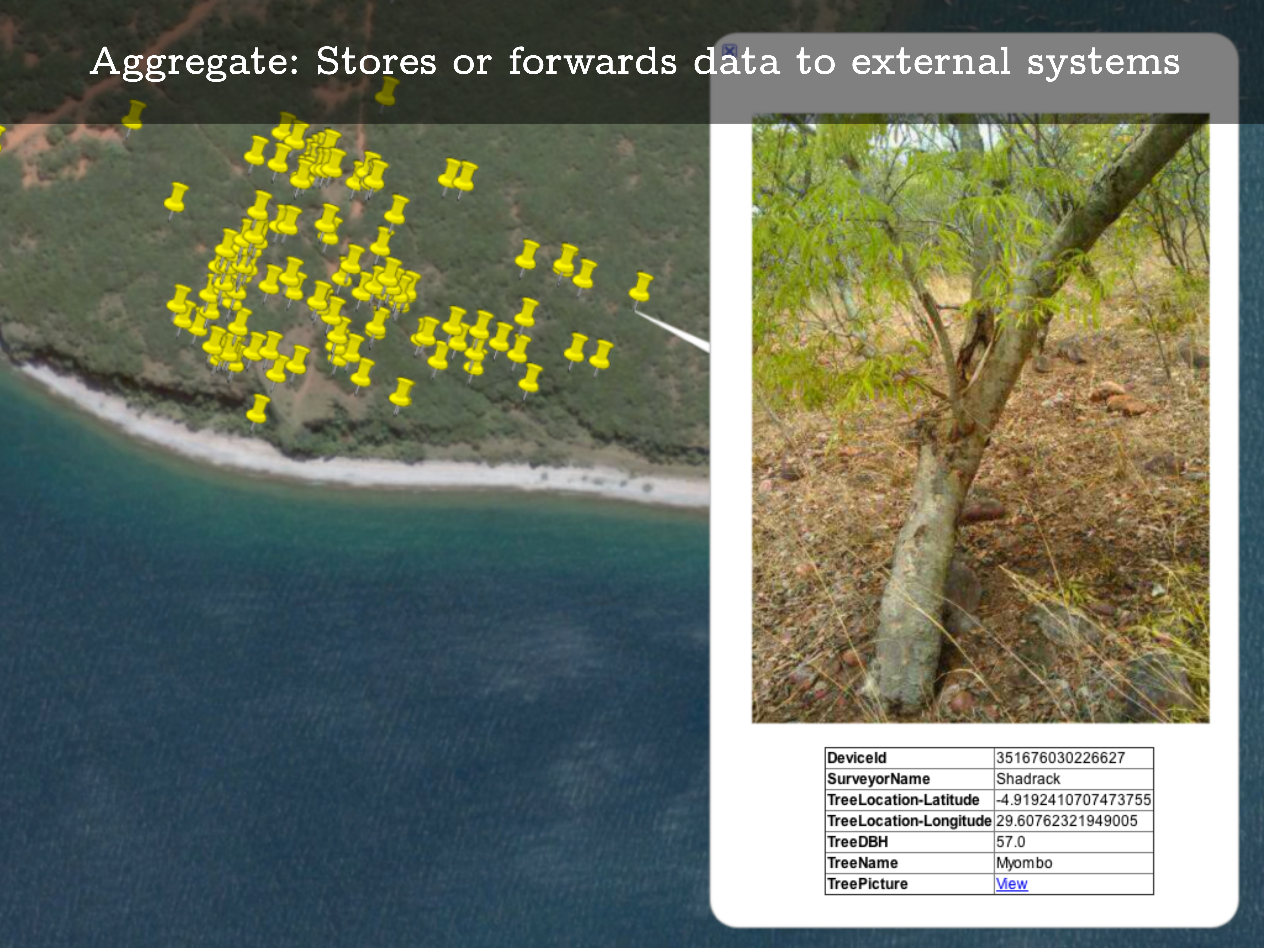
Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Comment
0	PRIMARY	1	db	A	NULL	NULL	NULL	
0	PRIMARY	2	name	A	0	NULL	NULL	

created: 8/18/10
updated: 8/18/10
rows: 0
size: 0 B
encoding: utf8

Another feature that makes Aggregate unique is that it designs the database backend for you using the XForm.

You don’t have to know how to use complex relational tools databases like these.

You give Aggregate the same form you give to Collect, and it builds a custom database for you.



Aggregate: Stores or forwards data to external systems

DeviceId	351676030226627
SurveyorName	Shadrack
TreeLocation-Latitude	-4.9192410707473755
TreeLocation-Longitude	29.60762321949005
TreeDBH	57.0
TreeName	Myombo
TreePicture	View

Finally, Aggregate can either store or forward your data to other systems.

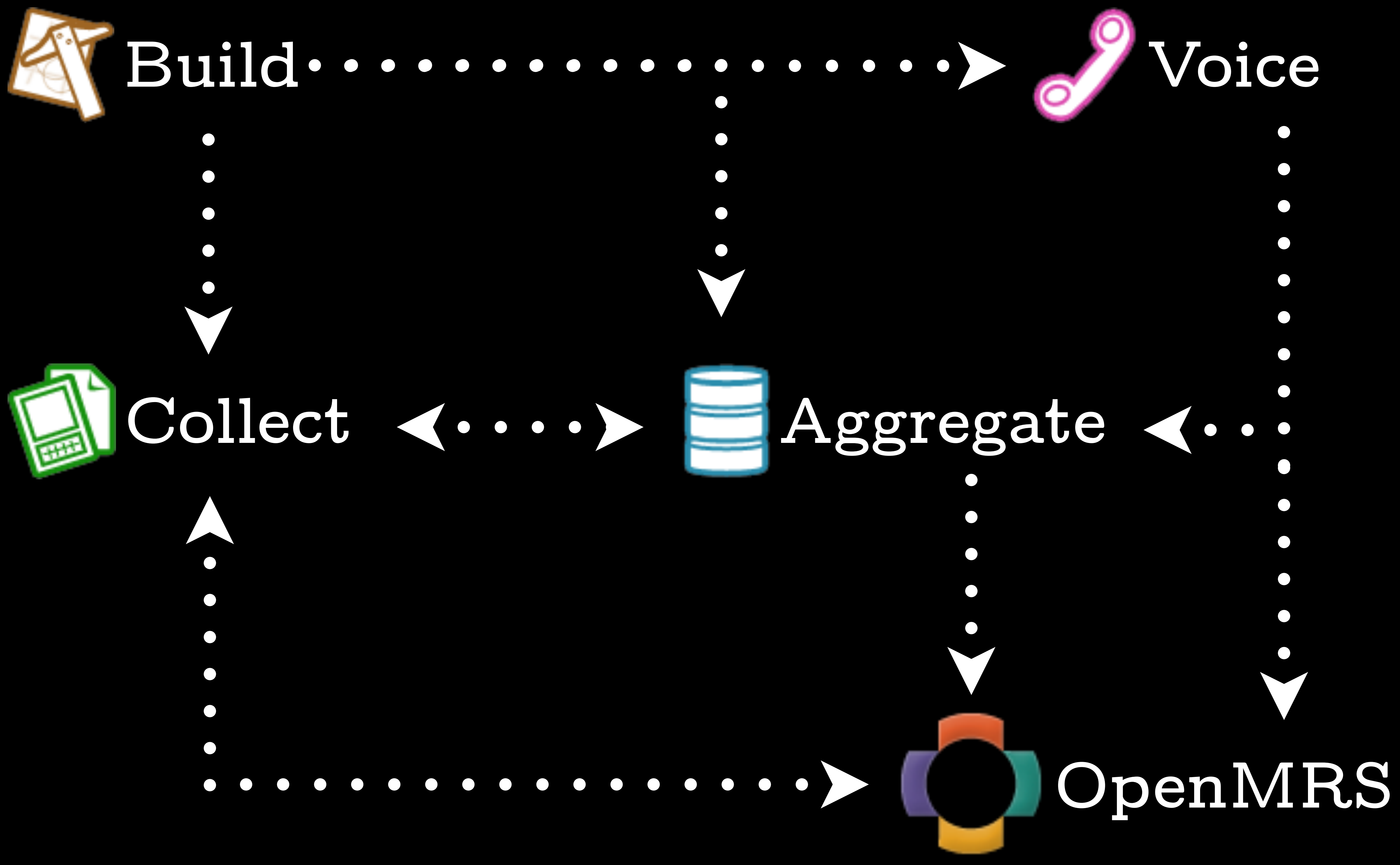
In this example, forestry workers with the Jane Goodall Institute in Tanzania, submitted data from Collect to Aggregate and then exported to Google Earth.

Managers could then click on each yellow point and get the data that was submitted.

The process of multilingual form design, training people who hadn’t seen smartphones, data collection and reporting took half a day.

So I’ve showed you three of the ODK tools, let me show how they fit together.

ODK Ecosystem: Tools designed to fit together



Build

Voice

Collect

Aggregate

OpenMRS

You use Build to design a form and you give that from to Collect to display it.

You give it to Aggregate, and it will build the storage for the responses.

To complete the loop, Collect can send data to Aggregate, and it can also download new forms from the Aggregate.

But ODK is more than those three tools.

For example, you can give the form to ODK Voice.

Voice is a tool we have that builds an IVR tree out of that form and connects it to the phone network. It does all the things Collect does with logic and multiple languages, but it does it using the voice channel on basic phones.

For example, you can have Voice call all the people in a district and ask them for their gender and language.

Based on their responses with their keypad, you can play the women pregnancy advice and the men farming advice. And it will do that in their native language.

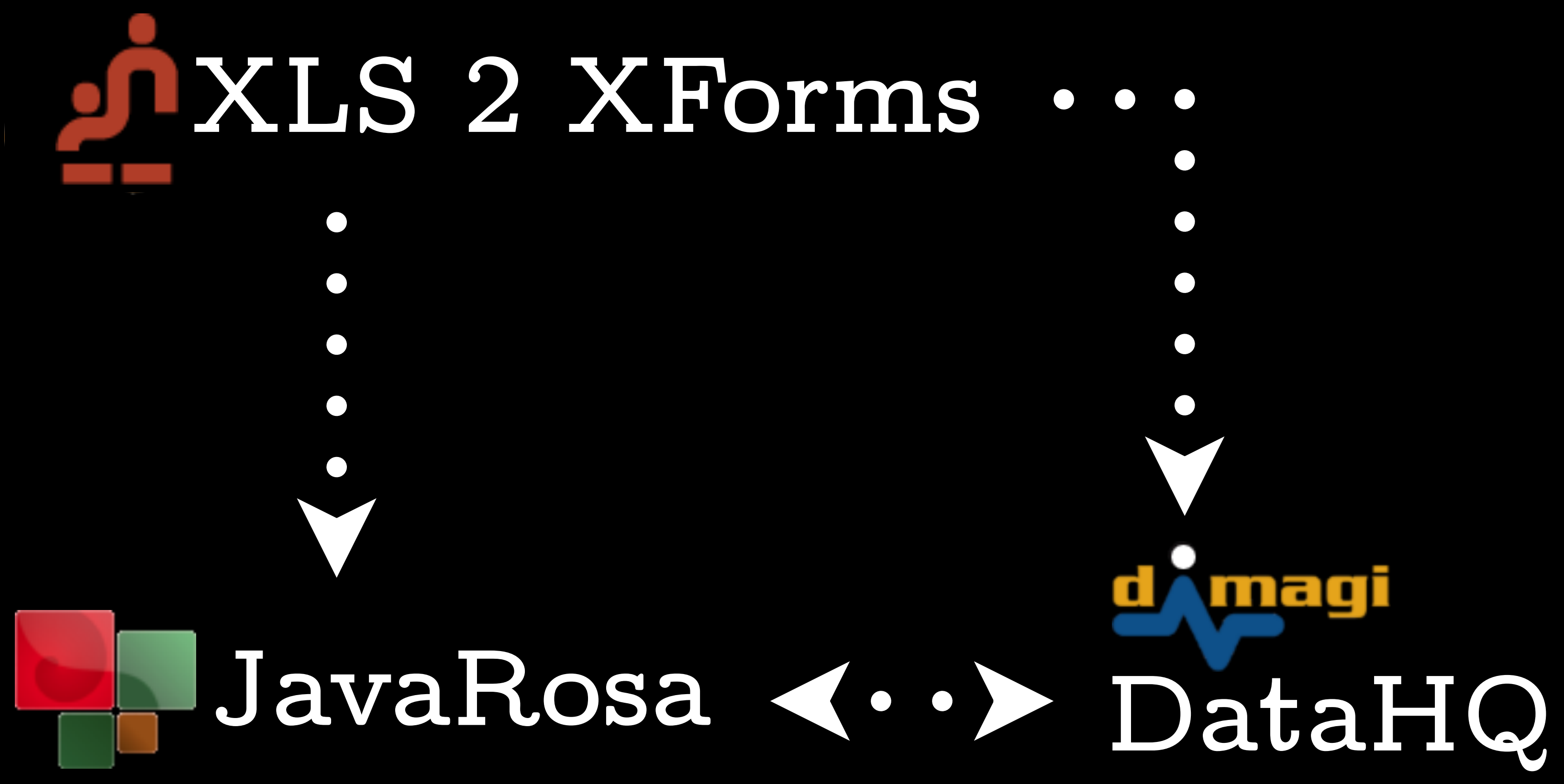
Naturally, all their responses get fed back to Aggregate.

We support connecting to other systems, say like OpenMRS.

You can treat OpenMRS just like Aggregate, and use it to share medical forms and data with Collect.

If you want your OpenMRS forms to be voice-enabled, you can connect Voice to it.

ODK Ecosystem: Design your system in about an hour



XLS 2 XForms

JavaRosa

dimagi

DataHQ

You might be asking how long it takes to build the basic end to end system.

If you are an implementer with basic computer skills you can be up and running with the form, the mobile client and the server in about hour.

More importantly, we are compatible with a bunch of other tools that may work better for organization.

If you have really large forms, you might want to use ChildCount’s XLS2XForms and that will work just fine.

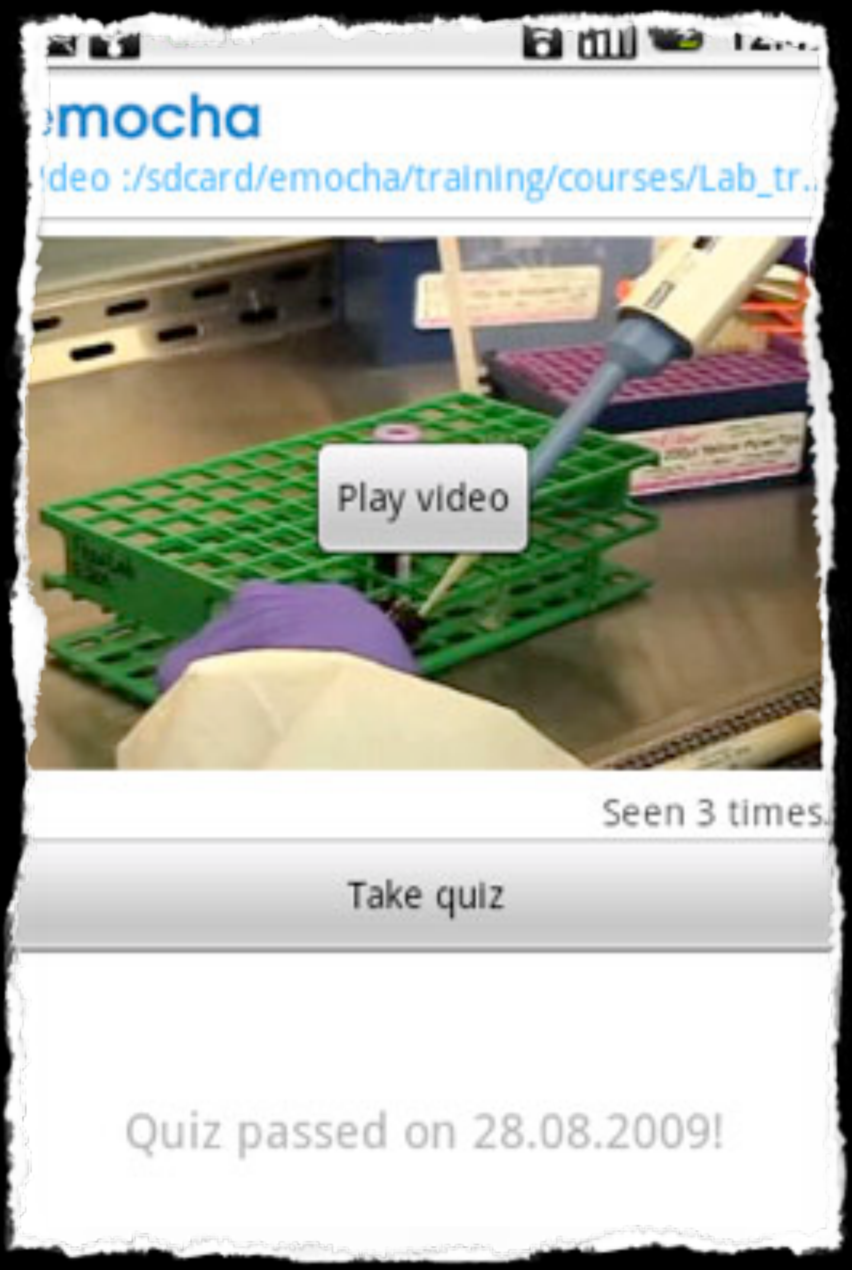
If you think Android phones are too fancy, you can use a J2ME client like JavaRosa. You loose the Android specific features, but everything will work.

If you don’t like our Java server, Dimagi makes an awesome Python one called DataHQ -- you can use that as well.

Now these are nice ideas, but like Ken said in the panel, does any of this actually work in the field?

We surveyed a few groups who were familiar with existing systems and had used ODK.

The details of the survey are in the paper, but I want to highlight what four of those groups said.



Seen 3 times.

Quiz passed on 28.08.2009!

“[ODK] is used for patient data gathering, and to give an exam after watching a video course on the phone.”

Johns Hopkins Center for CGHE

The first group said.

“[ODK] is used for patient data gathering, and to give an exam after watching a video course on the phone.”

This is from John’s Hopkins Center for Clinical Global Health Education.

They have their own tools, like a server that generates clinical statistics and a custom phone client that has lots of video courses.

They needed a way to test their users and so that’s what they use ODK for -- to take the test and send those results to their server.

They noted that when they started, they were concerned with relying on another group’s code.

Despite those concerns, they’ve been pretty happy with the experience.

In fact, their project was a finalist in the Vodafone’s Wireless Innovation Project.

“Features sometimes trump cost…we needed to get more out of the phone and ODK allowed us to do that.”

D-Tree International

The second group noted that

“Features sometimes trump cost...we needed to get more out of the phone and ODK allowed us to do that.”

This is from D-Tree

D-Tree is an NGO does electronic clinical decision support in Tanzania.

They’ve always used low cost devices like the ones shown here running JavaRosa. They have been trying to do more with the phones and ODK enables that.

What’s interesting is that they also use XForms, so the same JavaRosa infrastructure that worked on their basic phones works with ODK.

They can give the low-end phones to CHWs and give the high end ones to supervisors and adapt needed.

D-Tree was concerned with security issues of the smartphones being used for rural data collection. This is one of the reasons why they are trying it out mostly in facilities first.

“It is relatively easy to train using ODK...even for people who have little or no computer experience.”

Berkeley Human Rights Center

The third group said, “It is relatively easy to train enumerators using ODK...even for people who have little or no computer experience.”

This is from the Berkeley Human Rights Center.

HRC documents human rights issues with long surveys and most of it is done offline.

They have used paper and looked at other electronic systems before using ODK

The thing that is interesting here, is that they have contributed ODK training guides and translation of the tools to the community. They’ve also and spun-off their own compatible tools that are more suited for their needs.

“[Health workers felt] the system facilitated their home visits and allowed them to collect higher quality data. We actualized higher cost savings over our previous PDA-based system.”

USAID-AMPATH

The final group said

“Health workers felt the system facilitated their home visits, and allowed them to collect higher quality data. We actualized higher cost savings over our previous PDA-based system.”

This is from USAID-AMPATH.

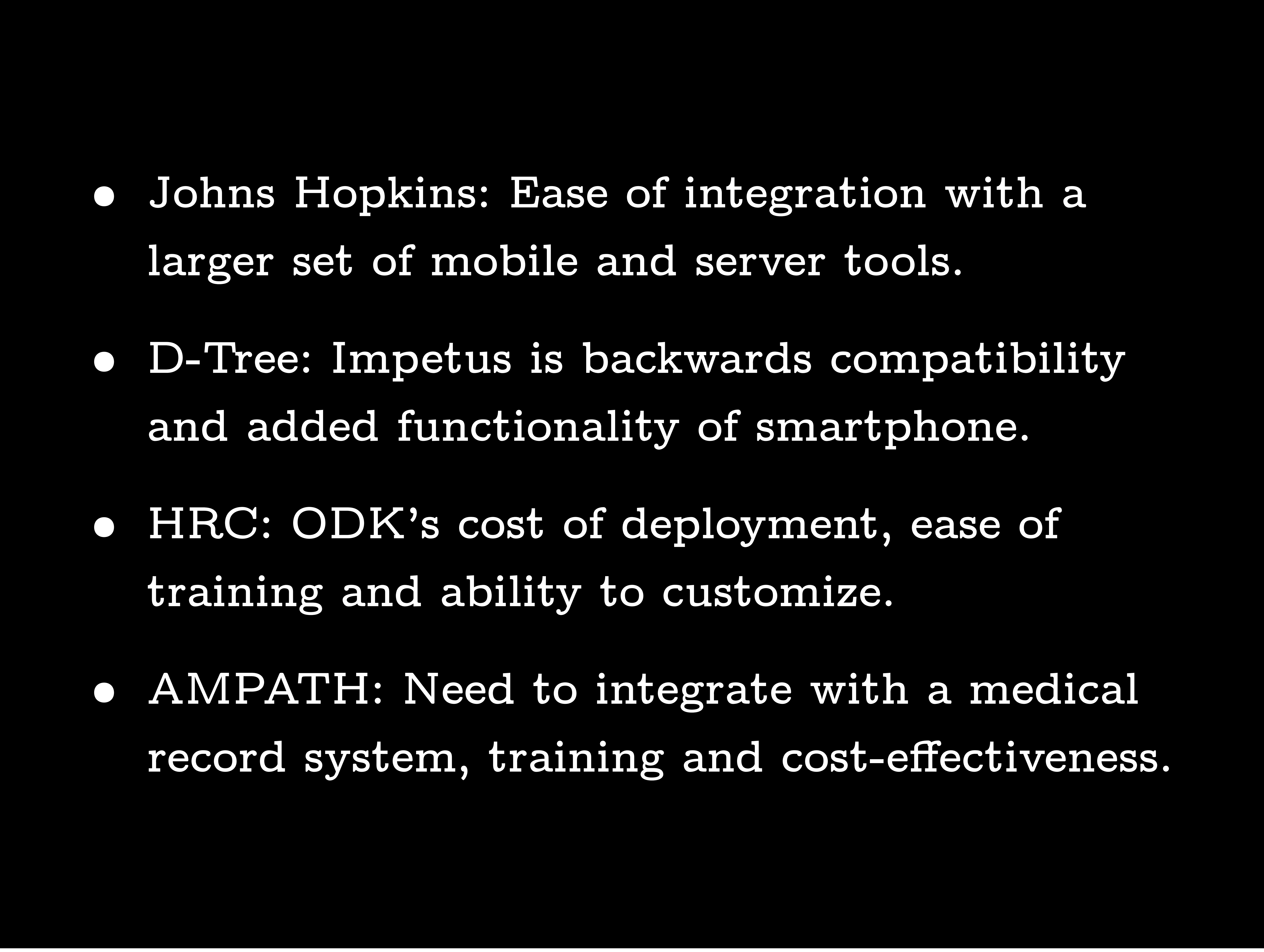
AMPATH is a very large hospital network in Kenya. They use ODK for home-based HIV counseling and testing of about 2 million people.

Previous to ODK, they used Pendragon Forms with PDAs and external GPS but found challenges with training.

It was also hard to integrate with their medical record system and that made reporting slow.

Using 200 phones, they’ve collected data on 63k people in 18k households and because of the early reporting that ODK enables, they can now act on critical issues they’ve found in their population.

So to summarize



- Johns Hopkins: Ease of integration with a larger set of mobile and server tools.
- D-Tree: Impetus is backwards compatibility and added functionality of smartphone.
- HRC: ODK’s cost of deployment, ease of training and ability to customize.
- AMPATH: Need to integrate with a medical record system, training and cost-effectiveness.

JHCCGHE: Ease of integration with a larger set of mobile and server tools.

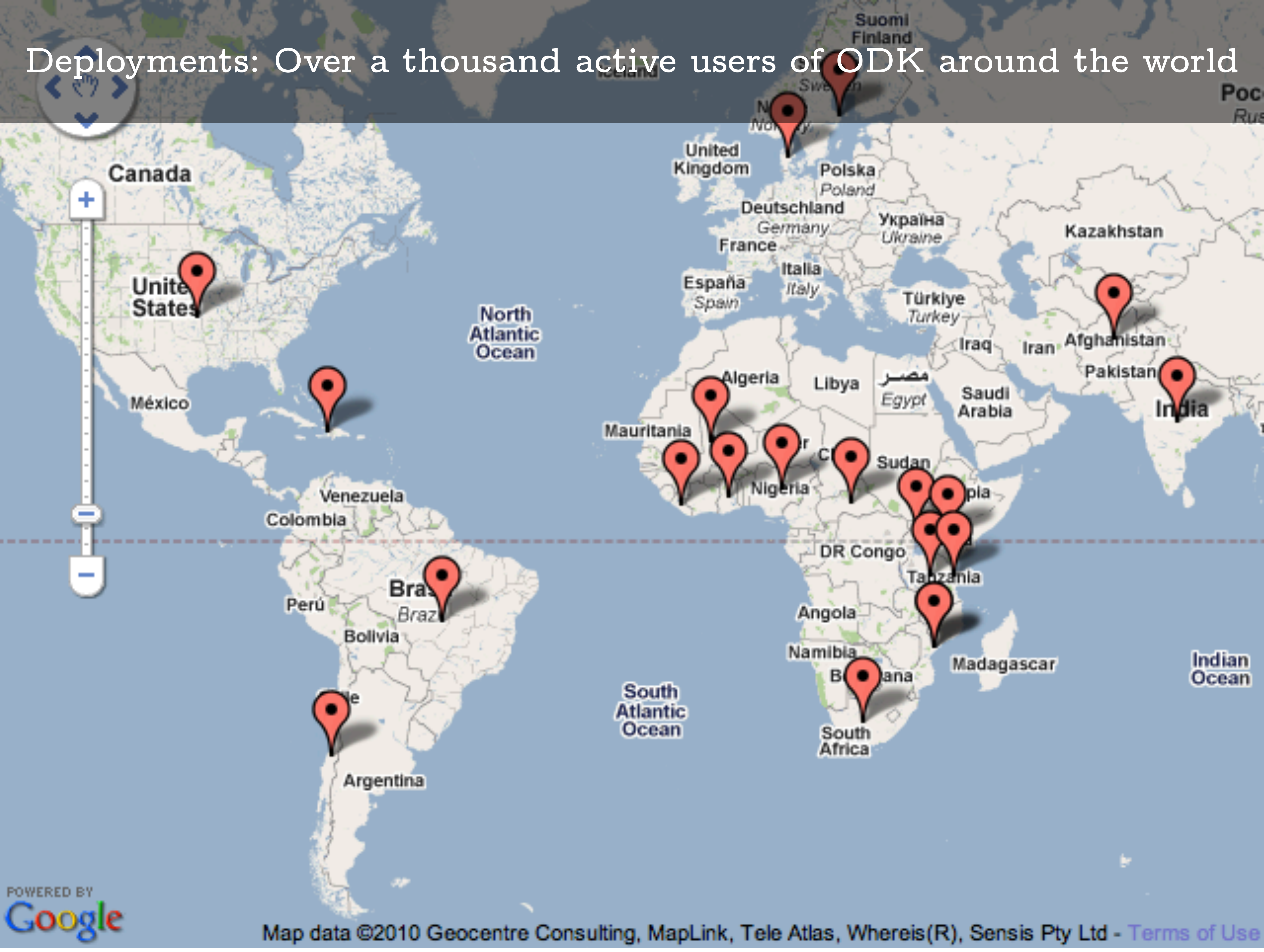
D-Tree: Impetus is backwards compatibility and added functionality.

HRC: ODK’s cost of deployment, training and support of open source community.

AMPATH: Need to integrate with a medical record system, training and cost-effectiveness.

Certainly the number of users we surveyed is small, but our downloads suggest

Deployments: Over a thousand active users of ODK around the world



Map data ©2010 Geocentre Consulting, MapLink, Tele Atlas, Whereis(R), Sensis Pty Ltd - Terms of Use

Over a thousand active users of ODK tools all over the world.

This map shows you just the countries where we’ve been told about projects.

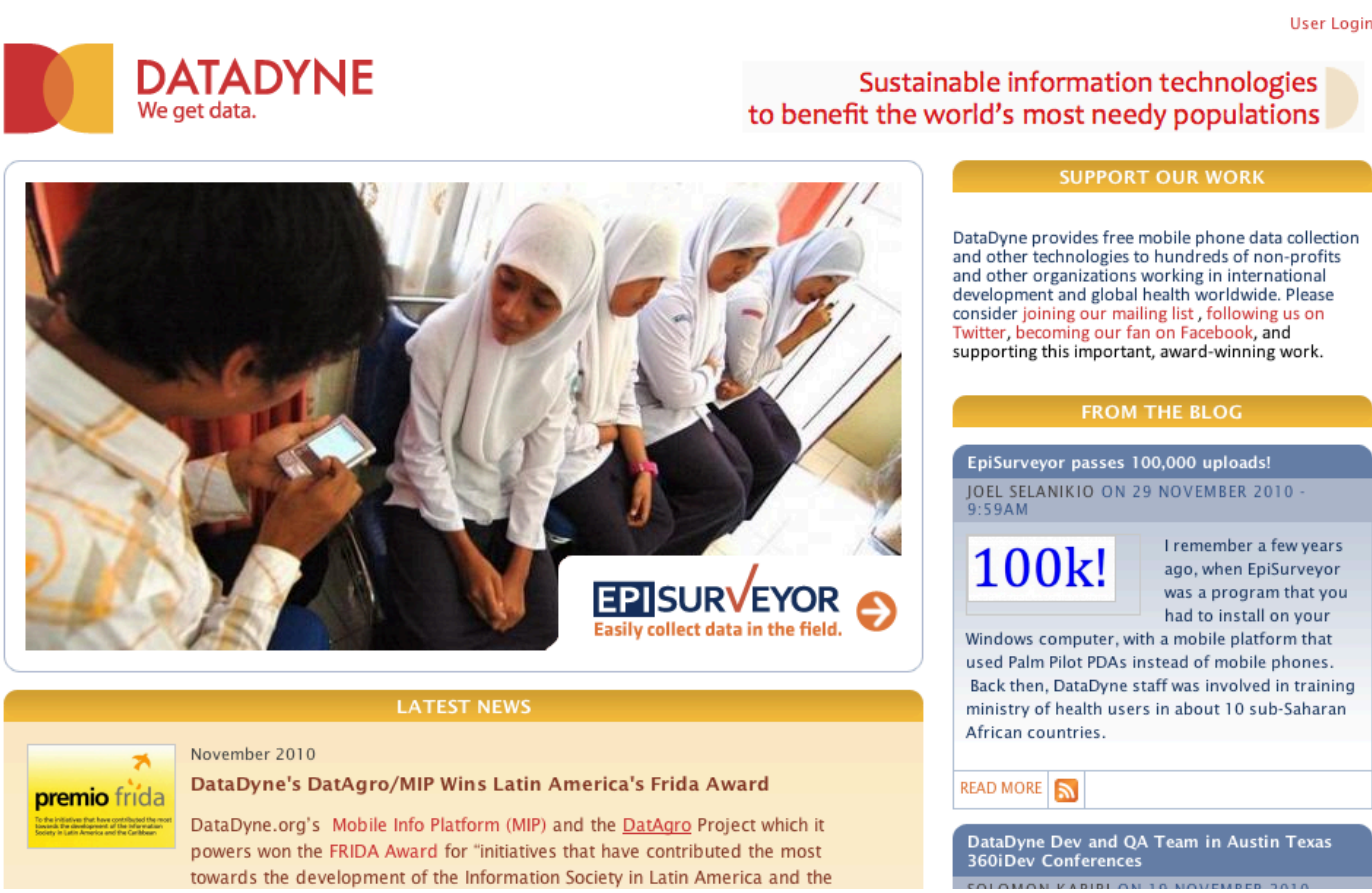
Some of these are small, like the farm in the US using ODK to collect plant soil samples in greenhouses.

Others projects are really large. The Amazon Conservation Team and Google are collecting and sharing lots of forest carbon data in Brazil with ODK.

Grameen Foundation in Uganda is using it in their CKW program.

There is a group in Mozambique who collected some 15k large surveys in 3 months.

Community: Individuals and organizations are contributing to ODK



User Login

DATADYNE
We get data.

Sustainable information technologies
to benefit the world's most needy populations

EPISURVEYOR
Easily collect data in the field.

SUPPORT OUR WORK

DataDyne provides free mobile phone data collection and other technologies to hundreds of non-profits and other organizations working in international development and global health worldwide. Please consider joining our mailing list , following us on Twitter, becoming our fan on Facebook, and supporting this important, award-winning work.

FROM THE BLOG

EpiSurveyor passes 100,000 uploads!

JOEL SELANIKIO ON 29 NOVEMBER 2010 - 9:59AM

100k!

I remember a few years ago, when EpiSurveyor was a program that you had to install on your Windows computer, with a mobile platform that used Palm Pilot PDAs instead of mobile phones. Back then, DataDyne staff was involved in training ministry of health users in about 10 sub-Saharan African countries.

READ MORE

DataDyne Dev and QA Team in Austin Texas 360iDev Conferences

SOLOMON KARIBI ON 10 NOVEMBER 2010

LATEST NEWS

premio frida

To the initiatives that have contributed the most towards the development of the Information Society in Latin America and the Caribbean

November 2010

DataDyne's DatAgro/MIP Wins Latin America's Frida Award

DataDyne.org's Mobile Info Platform (MIP) and the DatAgro Project which it powers won the FRIDA Award for "initiatives that have contributed the most towards the development of the Information Society in Latin America and the

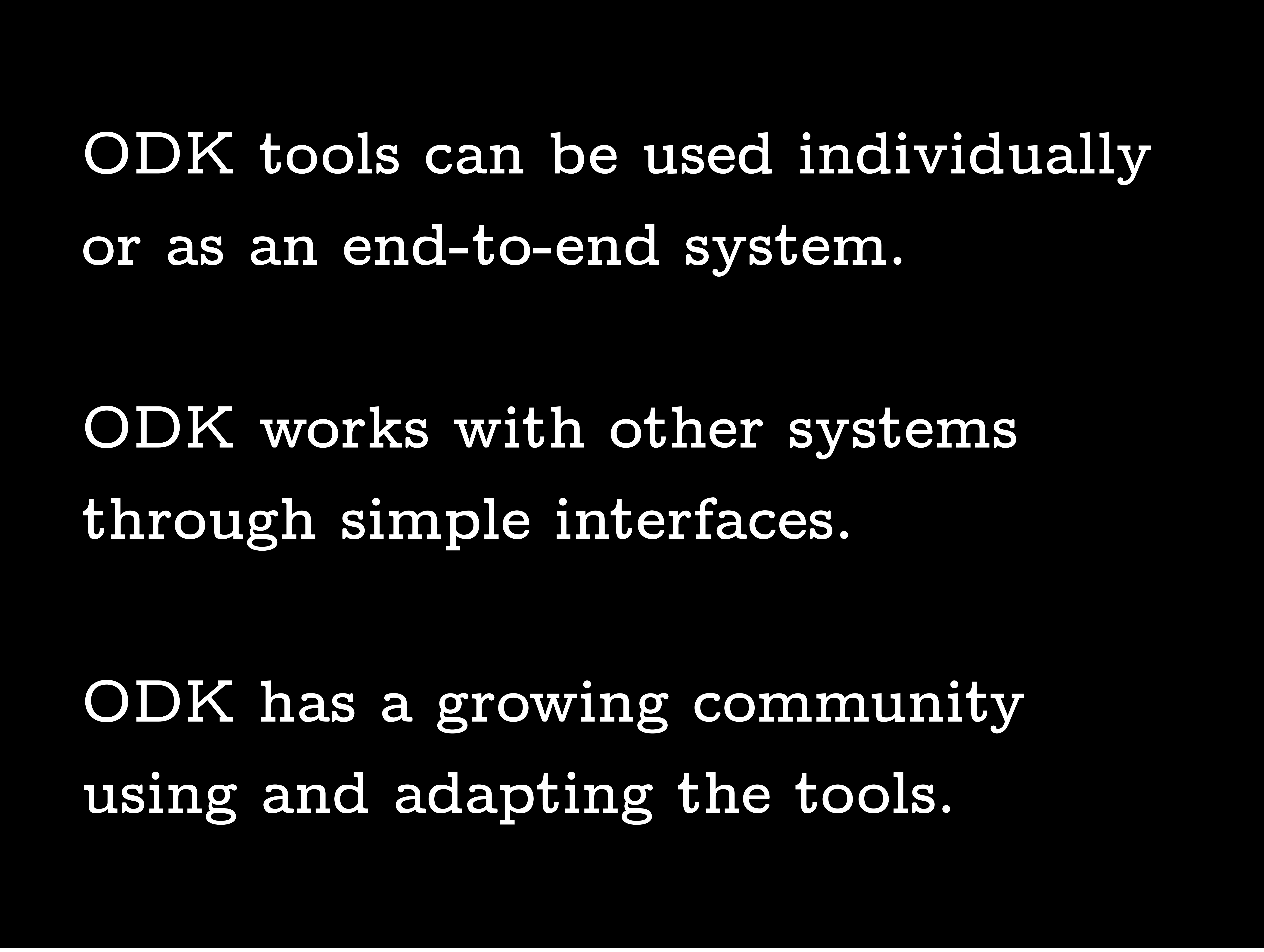
The community is also growing. We have seen independent developers helping, but we are also seeing companies making money building custom services with the platform.

Rhiza Labs sells ruggedized phones for ODK.

DataDyne, which makes the very popular Episurveyor tool, just released a Android client built from ODK Collect.

These anecdotes are not evidence, but they suggest we are on the right path.

In the paper, we conclude that ODK has demonstrated it can enhance information services in a variety of environments.



ODK tools can be used individually or as an end-to-end system.

ODK works with other systems through simple interfaces.

ODK has a growing community using and adapting the tools.

So to wrap up...

ODK tools can be used individually or as an end-to-end system for data collection or delivery.

ODK works with other systems through simple and asychronous interfaces.

ODK has a strong and growing community using and adapting the tools.

So with that, I do want to thank that community of developers, implementers and users and also want to give a shoutout

change

A group at the University of Washington exploring how technology can improve the lives of underserved populations.

http://change.washington.edu (@uwchange)

http://opendatakit.org (@opendatakit)

the Change group at UW

Are there are any questions I could answer?