DHIS2 and HISP
An overview

Johan Ivar Sæbø

Information Systems Research Group, IFI, UiO

# HISP and DHIS2

- Health Information Systems Programme (HISP)
  - A research/implementation project and network around health information systems in developing countries
  - University of Oslo, univerities in the South, various companies and individuals all over the world

- District Health Information Software 2 (DHIS2)
  - An open source software developed by HISP
  - Used in 60+ countries, some large NGOs

# Outline

- The problem
- The beginnings
- The philosophy
- The software
- The platform
- The development
- The use
- Demo

# The problem: To live healthy lives

- Good health is of value in itself
- Good health is a building block for "everything else": work, happiness, freedom, development

- Still, poor health services affects millions globally
- A «knowledge-do»-gap: we know how to improve health, but we fail to do it sufficiently

- The right information is crucial for making right decisions
- Appropriate technology?
  - Does what it is intended to do. What is intended? Evolution of needs
  - Infrastructure
- Technology is not a silver bullet. We build *systems*

# Health in "developing countries"

- DHIS2 is a software designed for and mostly used in the health sector in developing countries

- What is a developing country?
- What does it mean?
- What does it mean for us?

# A short side-story: Rødven Stave church



Photo: Frode Inge Helland, Wikimedia Commons

Oddleif Olavsen Rydjord

Born: Sept 2 1917

Died: June 5 1920

Gunnar Olavsen Rydjord

Born: Jan 1 1913

Died: Jan 20 1913

Kaare Olavsen Rydjord

Born: Dec 19 1911

Died: Jan 21 1912

Kaare Olavsen Rydjord

Born: July 18 1910

Died: July 18 1910

# Three lessons

- My grandfather would have a higher chance of survival if he was born in Ghana today (infant mortality rate approx 40) than in Norway in 1921 (infant mortality rate approx 60).

- Without access to medical services, it was/is not uncommon for parents to bury their young children

- But we know what to do! All over the world, we're moving away from the previous picture. How do we do that in an effective and efficient manner?

# «Developing country contexts»

- Large differences within countries. Urban-rural
- Income gaps, availability of health services
- Infrastructure. Internet, computers…
- Capacity: poor public institutions
- Capacity: to manage large complex information systems
- Capacity: digital proficiency
- Dependence on foreign aid, less exploited tax-base

- And, in some cases: extreme poverty, migration, war,
  – What you see in Norwegian news does exist, but is not the norm

# The implications of «developing country context»

- Need to be mindful of
  - Large differences in infrastructure, capacities, needs
  - Low bandwidth (an app can work perfectly in the capital)
  - Skills needed, both for use and for appropriation, development
  - End-users potential for self-support?
  - Licenses
  - Server management, prices, capacity
  - Routines and work practices
  - Etc etc

- Be mindful of Design-Reality gaps

# The beginnings

- HISP started in South Africa, with UiO involvement, 90s
- Extreme differences in health services
- DHIS1, 1.3, 1.4. Access based, desktop

- Action research
  - Learning while doing
  - Doing together with health staff

- Clear philosophy of how to approach the problem of:

  *Empowering local health staff with the right information, at the right time, to make the right decisions*

The International Conference on Primary Health Care at the Lenin Convention Center in Alma-Ata in September 1978.

# The philosophy - foundations

- Users know best – adaptability and participation

- Decentralization – support local adaptation

- «Primary health care»: health for all, preventive, «health district»

- Primary health service, majority of health services. Maternity, children, diseases
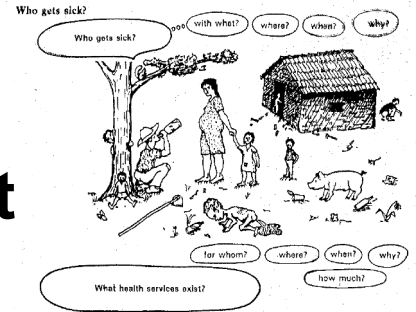
- Open source, open knowledge.

# The philosophy – software development

- Generic features: should be relevant across countries and use cases

- User input and feedback is important: but hard to manage with scale

- Free and evolving: but who will pay?

- Towards a platform
  - What is static can be in the core and API
  - What is dynamic can be in apps

# Philosophy: support Health Management

- Primary, Secondary, Tertiary services
- How many times have you needed health services?
  - As newborn: many times
  - As child: several times
  - As adult: when you're sick
  - If a woman: many times when you're pregnant

- Most health events are routine occurances
  - Pregnancies, immunizations, seasonal diseases

- Thus: DHIS2 focus on routine *monitoring and evaluation* and programme-specific *case management*

Picture: HMN

Picture: HMN

# The software



Typical Health Information Processes and Types

| Managerial | Administrative | Clinical |
|---|---|---|
| Performance data | Procurement | Test data |
| Service planning data | Contracting data | Diagnostic information |
| Demographic information | Resource utilization | Evidence-based medicine |
| Epidemiological statistics | Education and training | Care pathways and procedures |

- dhis2.org

- Support decisions in health

- Aggregate:
  - Are we on target? Do we immunize all children? Why not?

- Process:
  - When is your next visit? Which tests are you taking then? What are the results?
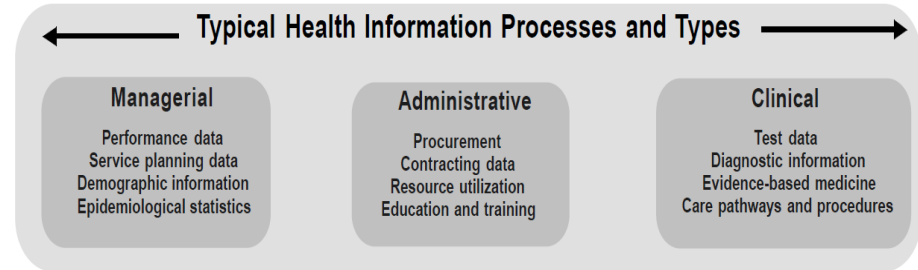
# The software – one part of the system

- DHIS2 as one, of perhaps many, applications
- Paper reporting still prevalent
- Need certain infrastructure
- Need a lot of skills

- Most of all, need a system of routines and work practices
  - All aspects of the information cycle

- Is embedded in at least one organization
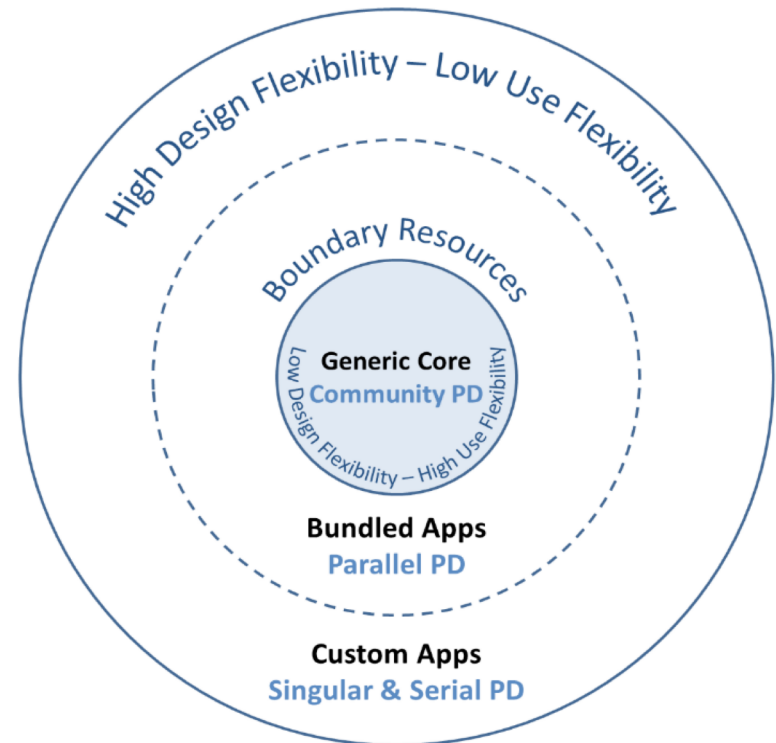  - Ways of doing things, assumptions

Collection

Processing

Presentation

Action
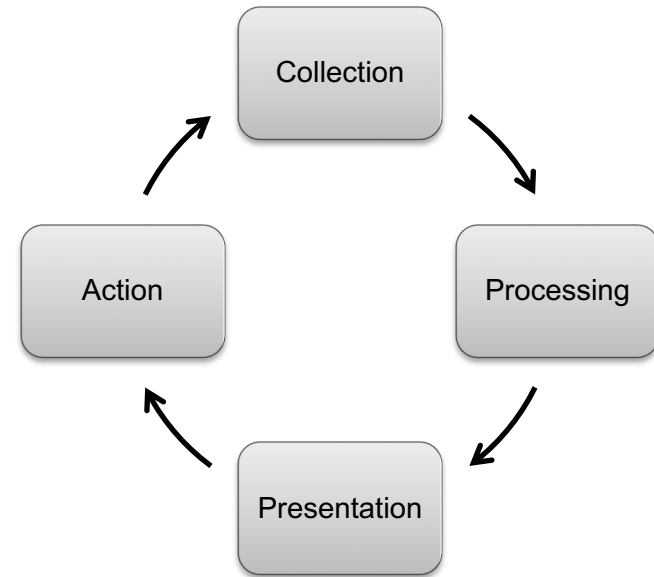
# The platform

- Why make a platform?

- Handle scale and complexity
- Foster innovation
- Generification

- Three parts:
  - Core (more stable)
  - API (more stable)
  - Apps (more dynamic)

# The platform – linked to philosophy

- The stable generic core:
  - Works for all
  - Limited user involvement

- Bundled apps:
  - Generic, made in-house
  - Somewhat more involvement

- Custom apps:
  - Free for all (like you)
  - High potential for user involvement



Roland, L. K., Sanner, T., Sæbø, J. I., & Monteiro, E. (2017). P for Platform. Architectures of large-scale participatory design. Scandinavian Journal of Information Systems, 29(2). Retrieved from http://aisel.aisnet.org/sjis/vol29/iss2/1

## Core and apps – the implications

- Most use cases share some common logic of how data is processed. Generic core can handle this, but needs to be stable

- But many use cases need specific things, perhaps not supported by any existing app

- WebAPI allows apps to use core. Innovation in specific use cases can be accommodated by building apps

- If new app is useful, it typically enters a phase of generification

# Core and Apps

- https://www.dhis2.org/downloads

- https://play.dhis2.org/appstore/

- https://play.google.com/store/search?q=dhis2

# The development

- A team at ifi – core, API, bundled apps
- Some development outsourced – special apps
- Some development by third-party developers (not coordinated by Oslo)

- The role of the users, changed over time
- Participatory design important, but hard with scale

**The use**
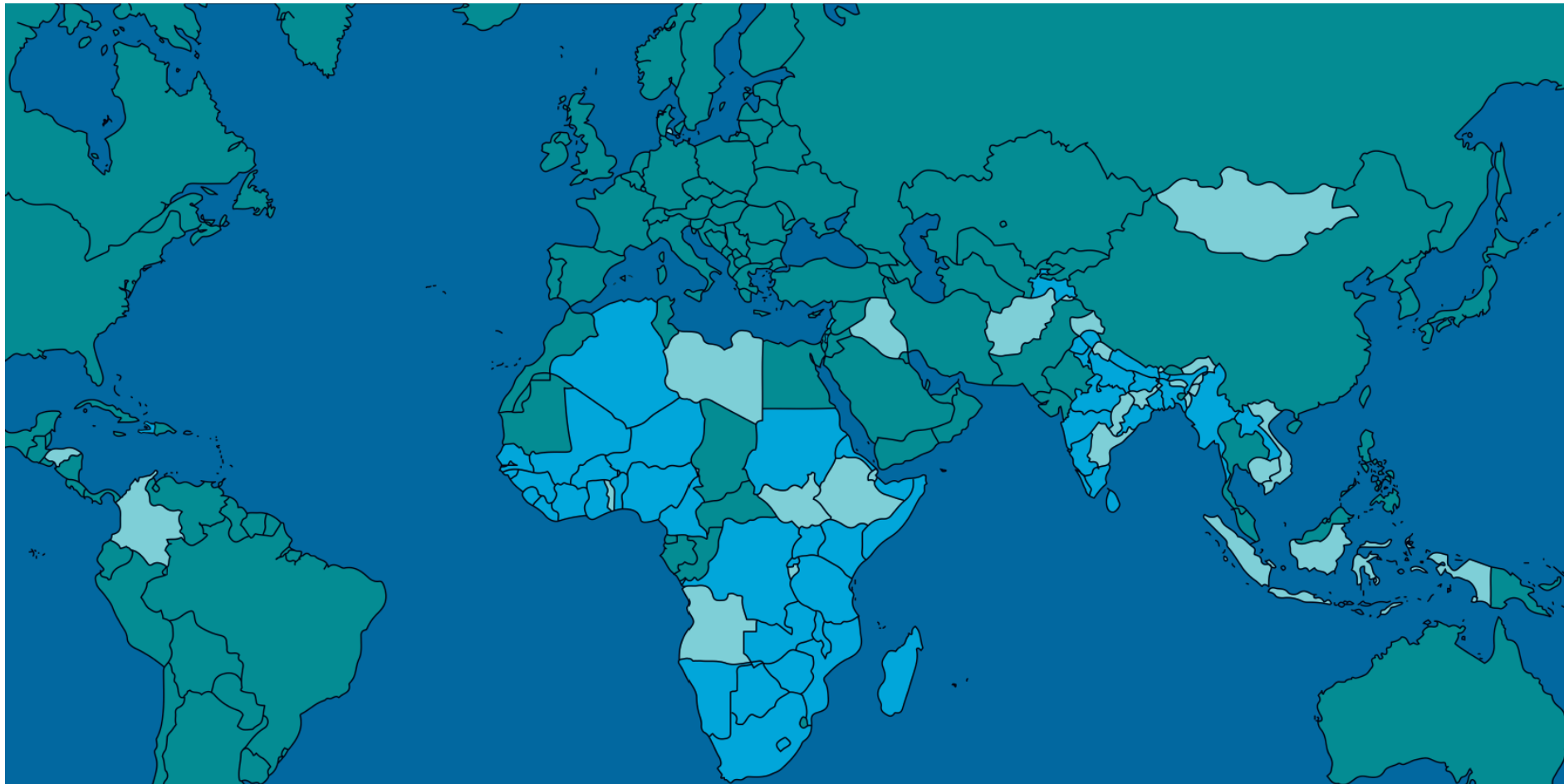
**Global footprint 2.28 billion people**

**+ 60 NGO's, 58 PEPFAR countries, 60+ PSI countries, 10 global organizations**
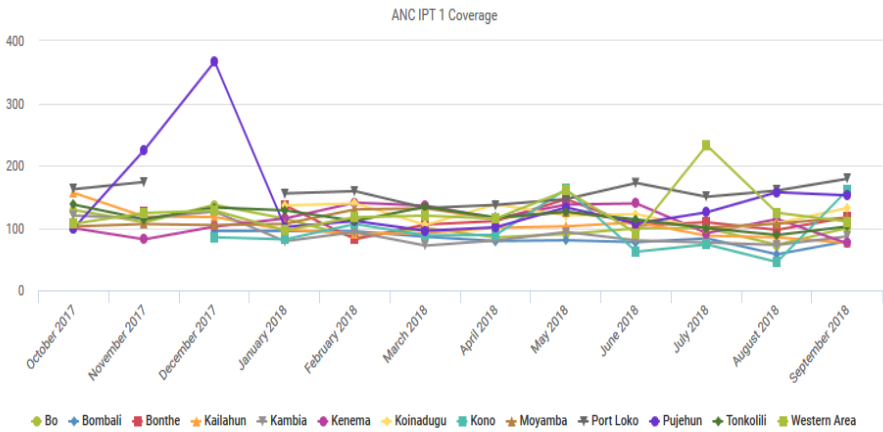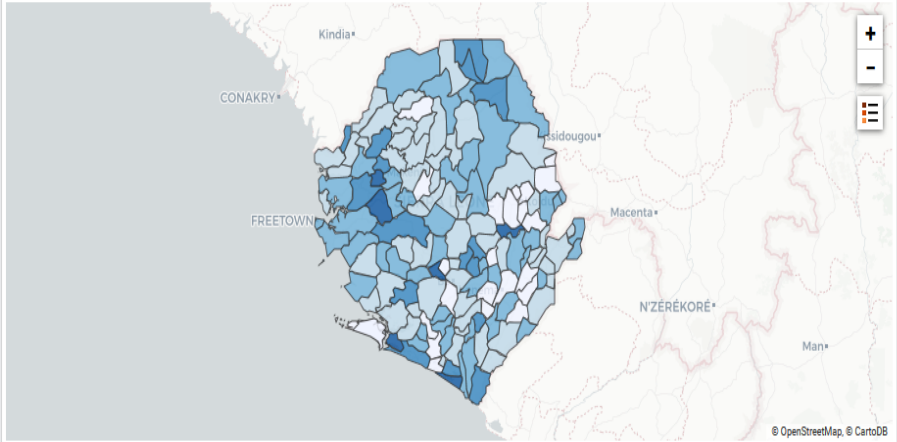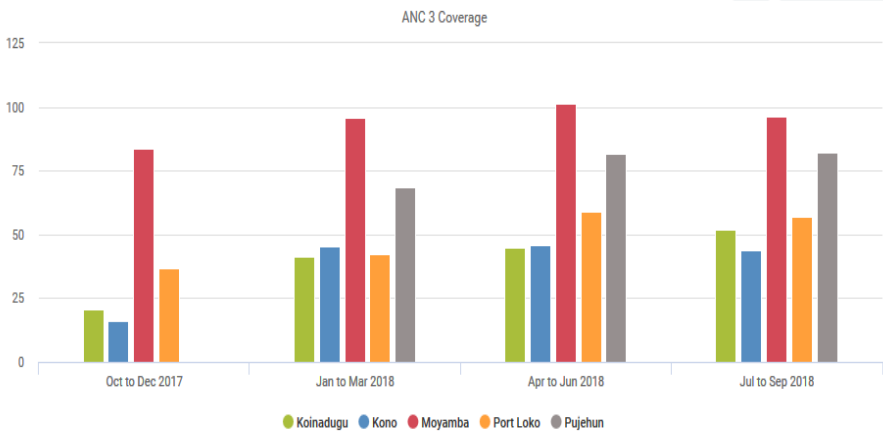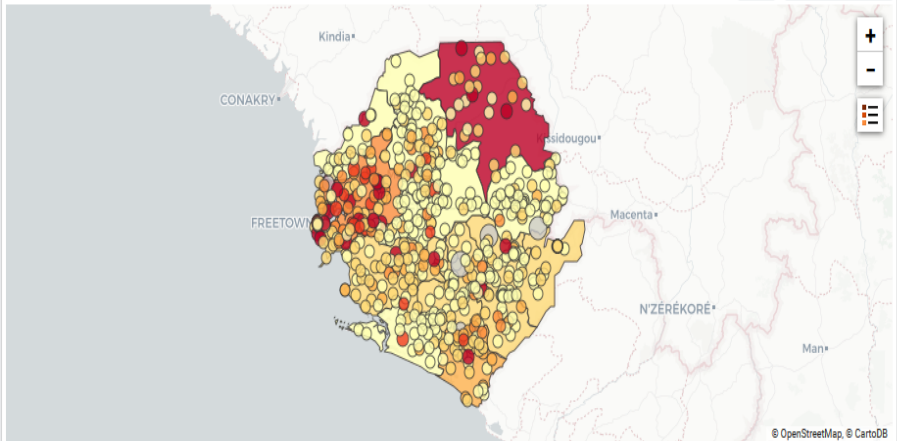
# Outline revisited

- The problem: to improve health service provision
- The beginnings: In South Africa in the 90s, small-scale
- The philosophy: FOSS, decentralized, participation
- The software: support decisions in primary health services
- The platform: ongoing process. Allows stability *and* innovation
- The development: mostly at ifi, also distributed
- The use: supports the information cycle in a range of countries
- Demo: up next

# Data logic

- What:
  - What are we measuring?
  - Hierarchy of building blocks: indicator – data element - disaggregations
- Where
  - All health events take place somewhere
  - A hierarchy of health service administration and provision
  - Organization units
- When
  - Fixed Periodicity (day, week, month, quarter, year, etc)
  - Point in time: more relevant for case based