COVAX Tracker and Analytics Performance Webinar

Friday, January 21 2022,

Agenda

1. Introduction - Mike
2. Server requirements and monitoring - Bob
3. Analytics solutions - Scott
4. Tracker solutions - Markus
5. Android solutions - Jaime B.
6. Implementation strategies - Rebecca
7. Tracker to aggregate data push - Olav
8. Q&A

Ask questions on CoP! Link provided in the zoom chat

Background

2021 saw a rapid expansion of large scale Tracker implementations, many related to covid vaccine programs, which share some challenging characteristics:

- Attempts to capture entire populations
  - New sites
  - High number of users
  - New users
- High requirements for real time data
- Significant negative impact of slow / down systems
- High usage throughout the day
- Importance of linked and longitudinal data

40 countries use DHIS2 for COVID vaccine delivery

dhis2

DHIS2 in action

DHIS2 is in use all over the world. Check out different use cases with this interactive map.

Health Information System

COVID-19 Surveillance

COVID-19 Vaccine

- Operational (40)
- In development (3)

Tracker

Packages / Tools:

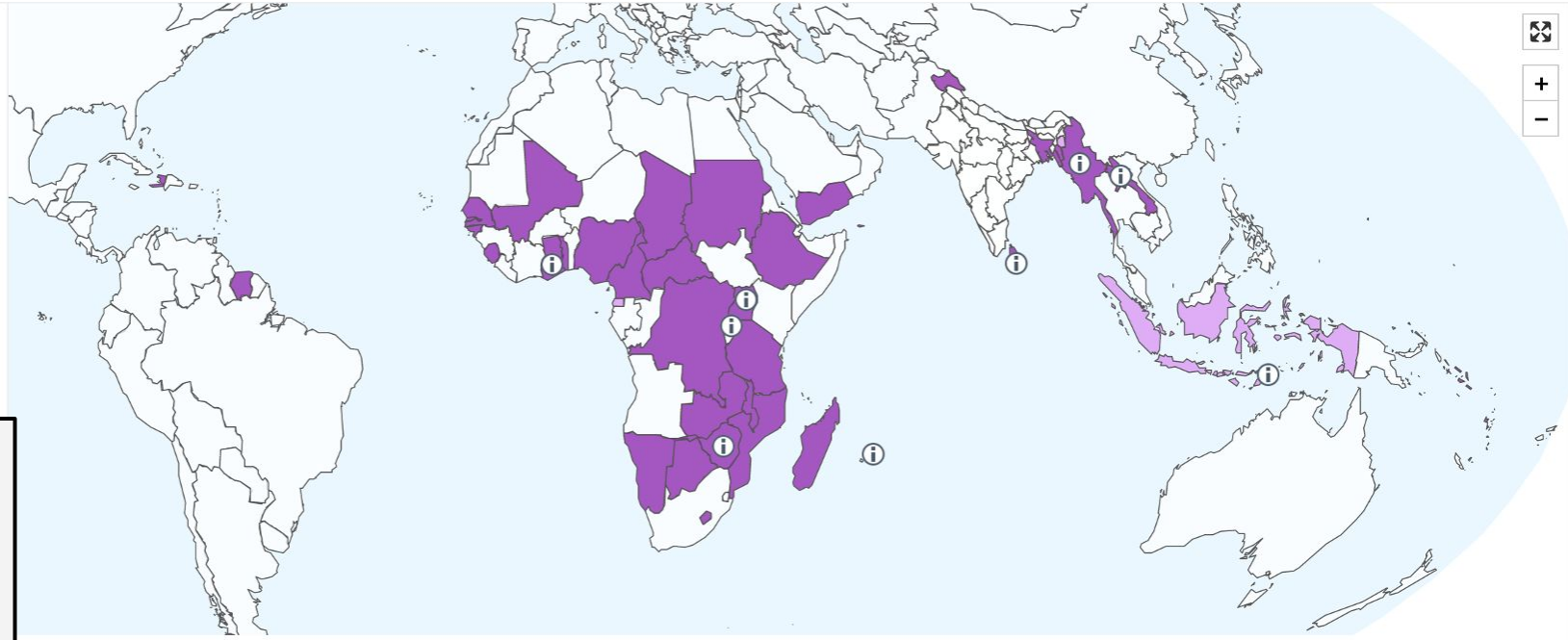
Electronic Immunization Registry (Tracker)

Core Analysis & Datasets (Aggregate)

AEFI (Tracker)

Logistics

Vaccine Certificates



Operational: 40

Bangladesh	Ethiopia	Malawi	Rwanda	Tanzania
Botswana	Ghana	Mali	Sao Tome	The Gambia
Cameroon	Guinea Bissau	Mauritius	Senegal	Togo
Cape Verde	Haiti	Mizoram	Sierra Leone	Uganda
Central African Rep.	Jammu and Kashmir	Mozambique	Solomon Islands	Vanuatu
Chad	Laos	Myanmar	Sri Lanka	Yemen
Dem. Rep. Congo	Lesotho	Namibia	Sudan	Zambia
East Timor	Madagascar	Nigeria	Suriname	Zimbabwe

In development: 3

Equatorial Guinea
Indonesia
Manipur

Covid Surveillance and Vaccine Systems at Scale

People Enrolled:

- **Sri Lanka**: 19,147,151
- **Rwanda:** 10,000,000+
- **Nigeria:** 7,913,042
- **Uganda:** 6,530,933
- **Ghana:** 3,633,623
- **Laos:** 2,456,865
- **Togo:** 1,751,575
- **Tanzania:** 1,107,619

Users

- **Nigeria**: 20,715
- **Tanzania:** 15,478
- **Rwanda:** 8000+
- **Togo:** 5,780
- **Sri Lanka:** 3,400
- **Uganda**: 2,810

Sites

- **Uganda:** 2,605
- **Rwanda:** 1,800
- **Togo:** 1,004
- **Sri Lanka:** 837

Covid-19 Related Events

- **Sri Lanka:** 25,863,786
- **Uganda:** 7,819,206
- **Laos:** 6,743,808
- **Ghana:** 4,525,188
- **Tanzania:** 2,110,405
- **Togo:** 1,410,691

As of 14 Dec 2021

Background

The recommendations in this webinar are based on lessons learned supporting some of the largest and most demanding Tracker implementations.

- These recommendations are aimed at system owners and sysadmins. They are technical in nature and require strong familiarity with your DHIS2 implementation.
- These recommendations **should be implemented from the start** of your rollout, not waiting for problems to arise

We will continue to learn from the challenges that you are facing, and the problems identified are being addressed through software development.

These **recommendations are likely to be updated** throughout 2022. They have been added as official documentation here: https://docs.dhis2.org/en/implement/tracker-implementation/tracker-performance-at-scale.html

Server and hosting requirements

The most important requirement is human!!!!!

You shouldn't expect to host a large scale tracker implementation without strong technical skills (we should make a list)

You might need to recruit and/or train

Hosting so much personal data is a security and privacy challenge. You need to have someone responsible for security and a basic security plan in place which complies to minimal good practice.

Server and hosting requirements

Server specifications will vary depending on scale. Most current deployments are using more than

- 32 CPUs
- 32 GB RAM
- SSD/fast disk

deployed to one or more virtual machines.

A good approach is to estimate the number of TEIs, orgUnits and users and find out what others of a similar scale are using

Be prepared to monitor (more later) and adjust resources according to how your own load develops

Shared environments

It is rare now to use fully owned physical hardware. Generally you will use one or more virtual machines from either a commercial provider or a national data centre.

For large high performance applications purchase dedicated resources rather than shared

(One big VM broke into containers can outperform many small VMs in a shared environment)

If working with local data centre (usually vmware), be prepared to test service level agreements and present evidence back to the provider regarding resource allocation. Over-provisioning and throttling resources is very common.

Software versions

Using the right versions of key software/packages is important for performance:

- JDK 11
- PostgreSQL 12 or 13
- DHIS2 2.35 or above - **latest patch release**

Several performance improvements have been made in the 2.35+ patch releases

Have a well rehearsed plan for testing and deploying new patch releases rapidly

Server monitoring

A DHIS2 environment (not necessarily just a “server”) is a complicated mix of tasks and resources.

How do we know:

1. Do I have enough (or too much) resources?
2. Did the performance intervention I just tried make a difference? How much?
3. What is struggling? The tomcat or the database?
4. How does my load today compare with yesterday, last week, last month
5. Which web requests are bringing down my server?
6. Is my disk performing as it should?

We can’t know the answer to these and other questions without measuring. Running a large tracker server without good metrics is like driving in the dark.

Simple tools have their place

htop to see current cpu usage
systemd-cgtop showing breakdown by container
psql to run queries on postgresql statistics tables (examples?)
grep, awk etc for analyzing log files

Tools which measure and record and display historical data give you a better overall picture of what is happening.

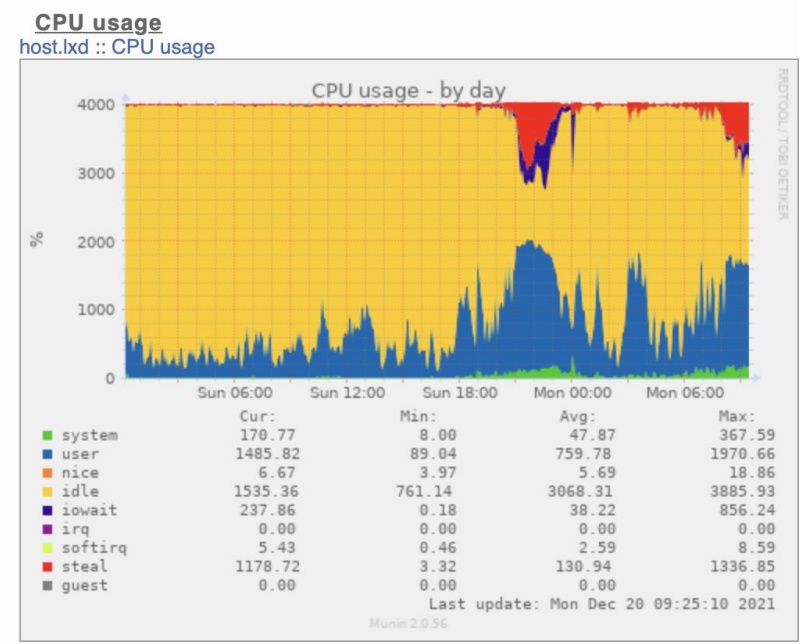
Prometheus+grafana, ELK stack, netdata are all popular.

Munin is not so pretty, but really easy to install and lots of useful metrics
Glowroot also easy to install and gives detailed insight into the jvm

Something wrong here?

The red stuff (steal) usually happens in overprovisioned shared environments.
You think you have 40 cpus, but can never actually use more than 20 :-(

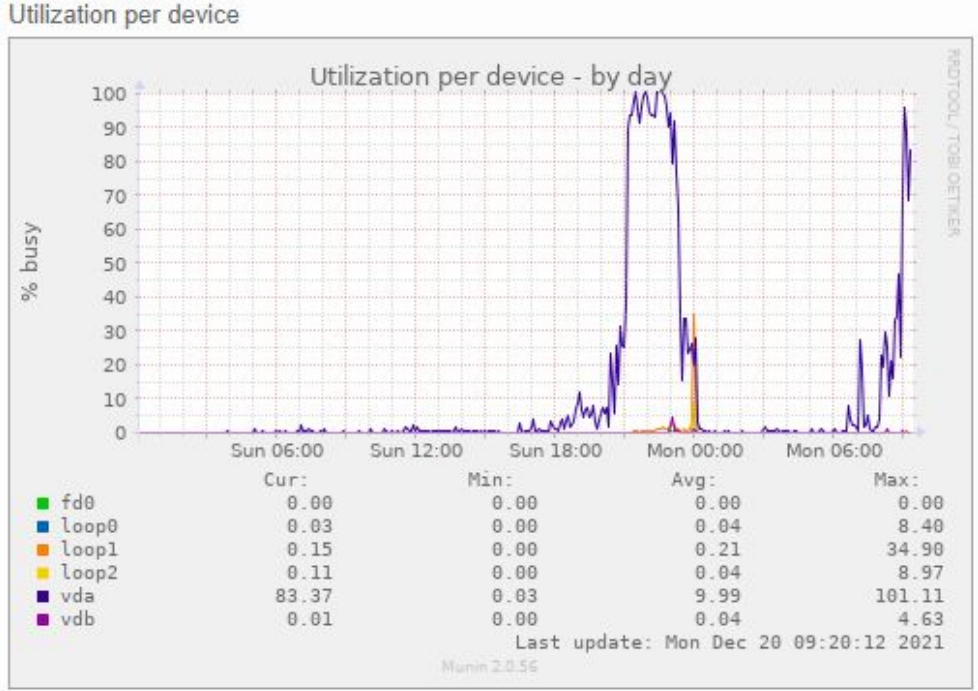
The purple stuff spells danger - CPU is spending a long time waiting for disk operations



Yes that disk is struggling

Usually this is either a slow disk or over-provisioning in a shared environment

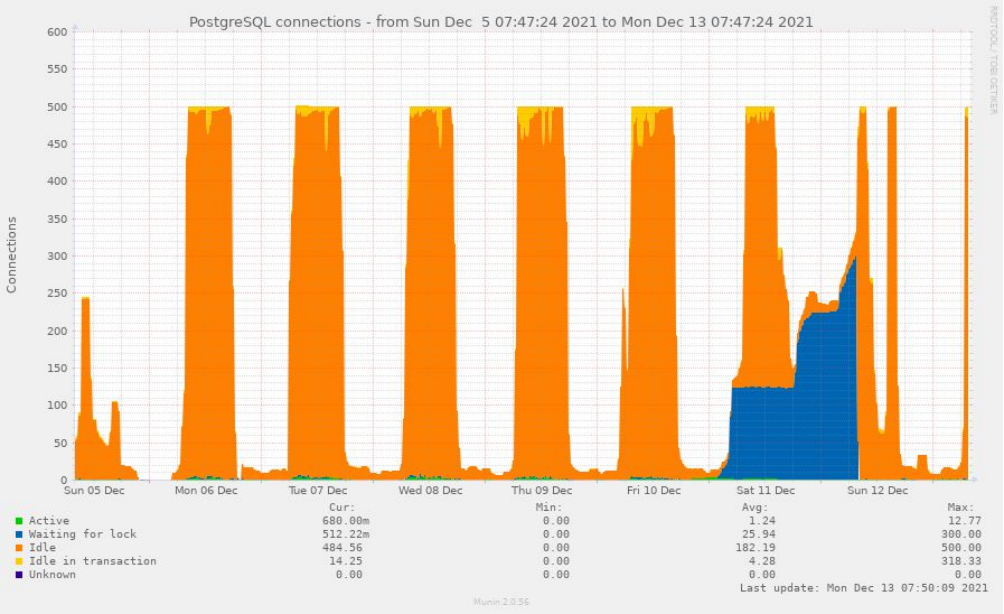
(disk latency is the other good metric to look at)



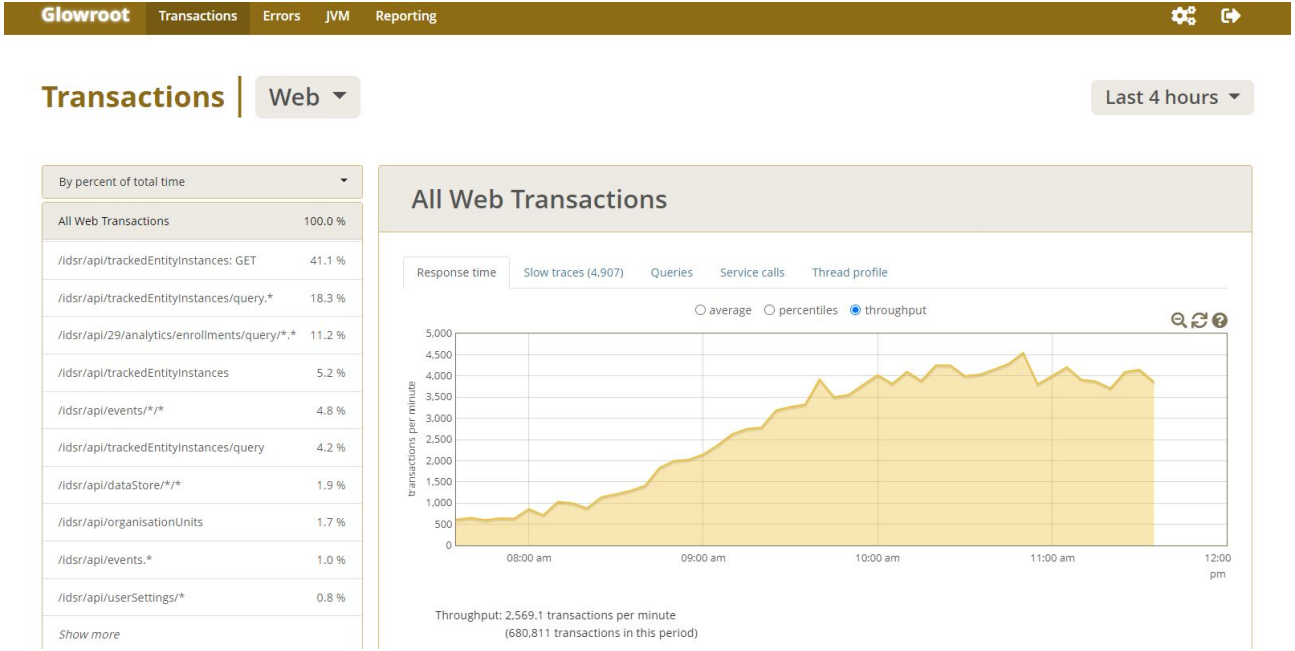
Something bad happened with my postgresql

Looking at the active postgresql connections gives a good picture of overall health

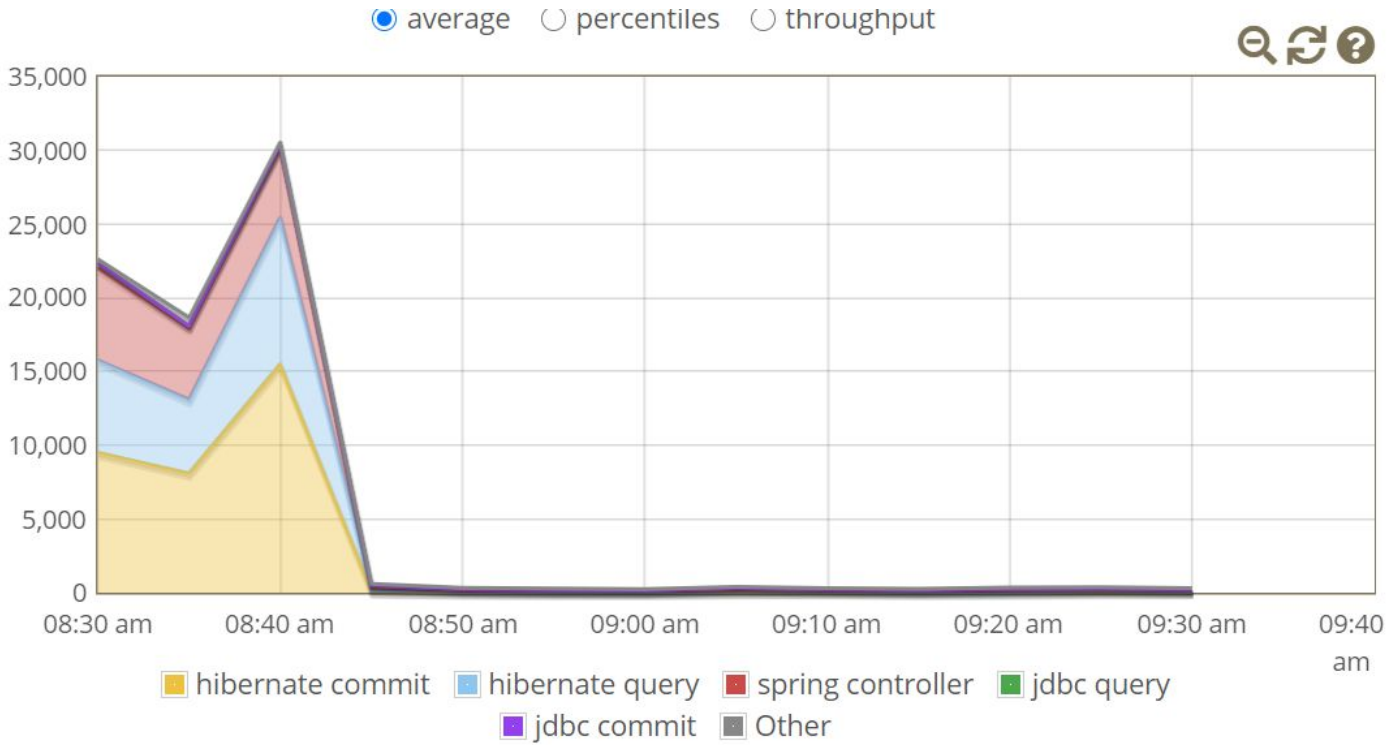
Something bad happened on Saturday which needs to be investigated further (probably a long running query which blocked other transactions)



Identify which API call is causing pain



The new war file has made a big difference



Scale and complexity requires monitoring

You need to know when you are in trouble
Rapidly zero in on where the pain is coming from
Be able to see whether your interventions are making a difference
Report useful evidence to developers as well as hosting providers

It is one thing to generate pretty graphs. The art is to interpret them.

Very often it is not possible, legal or appropriate to give developers access to the national database. But giving read access to web-based monitoring tools is a fantastic way of getting technical support.

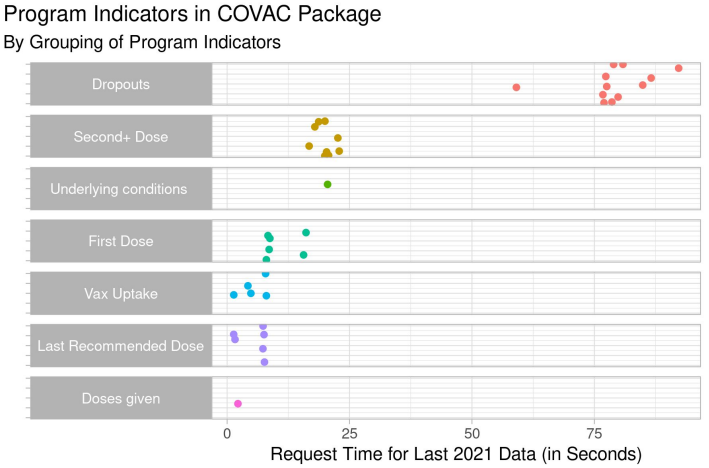
Analytics Problems

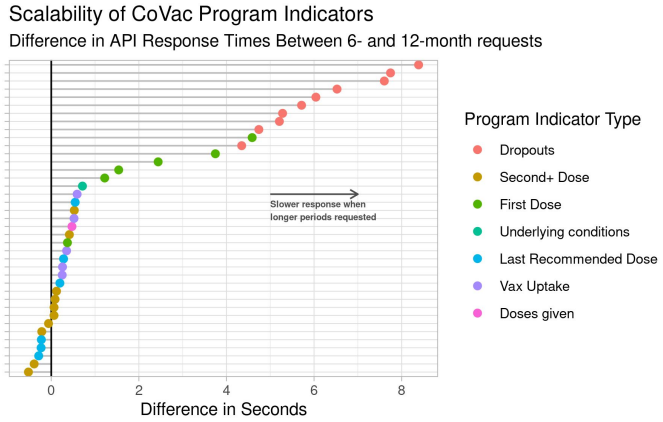
What is the problem

1. Slow performing dashboards or dashboard never loading
2. Line lists, tables, and charts taking a long time to load or never loading
3. Analytics tables failing
4. Worst case: servers crashing due to overloaded analytics queries.

What is causing the problem:

- Large scale tracker implementations (>500,000 - 1M tracked entities) with inadequate server specs or expertise.
- Heavy analytics requests
  - Maps at lower levels, or requesting unnecessary org units
  - Event reports with more than 100 event rows or 50 enrollment rows
  - Visualizations requesting lengthy periods of longitudinal data, e.g. last 12 months
  - **Any visualization with enrollment-type program indicators**, such as dropout rates
    - In tests of program indicators from the COVAC package, program indicators of “enrollment type” took the longest response time. Further, they have poorer scalability, as they take longer to serve data when requesting additional periods, org units, or TEI.



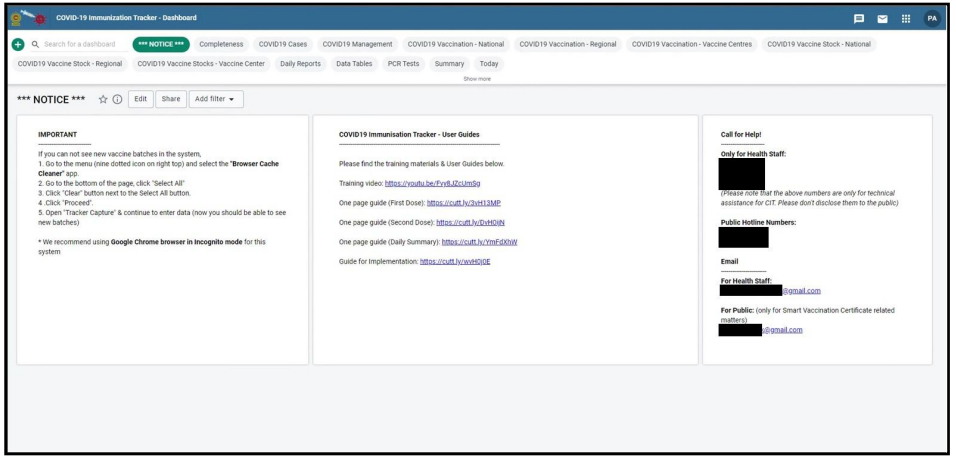


Testing on Performance instance v2.37 (3Mil+ TEI)

Analytics Solutions

What are some solutions:

- Ensure you have adequate server resources and support. Perform the server checklist.
- Do not run analytics tables during high usage periods during the day. Only run analytics tables at night.
- Use event type program indicators as much as possible. Enrollment counts can be very resource intensive.
- Do not have dashboards with program indicators set as the default landing page after log-in
    - Have an informational dashboard as the landing page after log-in.
    - Set the dashboard as public.
    - Star the dashboard
    - Name the dashboard **NOTICE** or **INFO** so that it shows as the first stared dashboard



Analytics Solutions

What are some solutions:

- Limit sharing of dashboard with program indicators to only those user that need to use the information .
  - Restrict access for data entry staff that do not need to see dashboards for work.
  - Restrict org units on the dashboard to only those that are necessary for each user. Use user relative org unit assignment for dashboard analytics.
- Tracker analytics requests, in particular for certain program indicator configurations, can be slow and create performance problems. When extracting such data:
  - Do it outside peak hours for vaccinators, to avoid any slowdowns hindering their work
  - Work with smaller datasets at the time. For example, it may be necessary to get figures for a subset of organisation units at the time (e.g. by region).
  - Rather than several people downloading the same data (e.g. for the national level) from DHIS2, download once and share via for example excel.

Analytics Solutions

What are some solutions:

- Ensure caching is enabled in the dhis2 configuration, so that repeated requests for the same analytics resources are served from the cache and database queries are skipped.
  - dhis.conf - analytics.cache.expiration set to at least 3600 (6-10 hour).
  - System settings -> analytics -> cache strategy. Recommended value: at least CACHE_6AM_TOMORROW. Set cacheability to “private” to avoid nginx cache.
- Turn-off continuous analytics. If you turn off continuous analytics you will only see your analytics updated after your analytics tables run.

Analytics Solutions

Last resort solutions - Nothing else has worked:

- Remove tracker analytics access for all non-critical users.
- Consider using SQL views, HTML reports, or R-shiny apps for analysis to product some program indicators that take too long to load.
- Push data routinely to a separate instance on a different server that use used only to product analytics.
- Set default landing-application for all users to data entry.

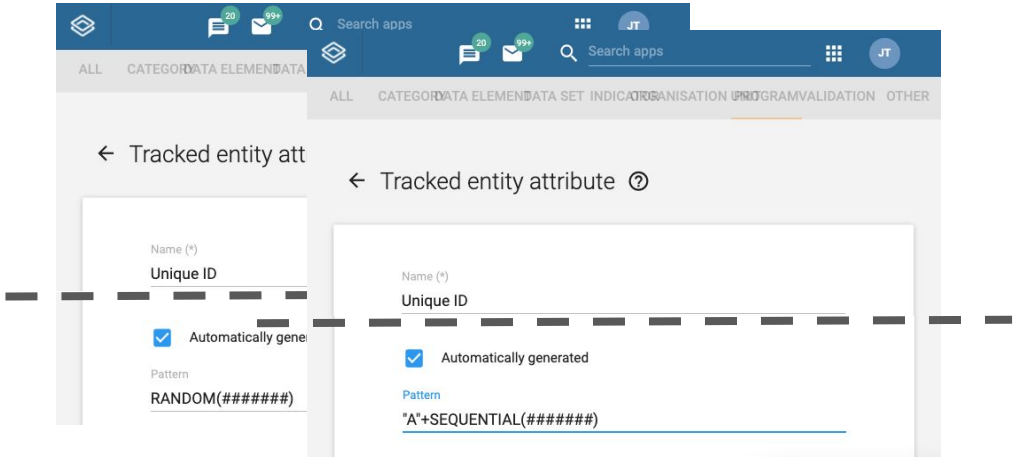
- Communicate with use when you experience serious analytics performance issues and what you have done to address it.

Tracker Solutions

6 hard earned learnings from the field that should be considered for all high volume DHIS2 instances

Tracker - ID Generation

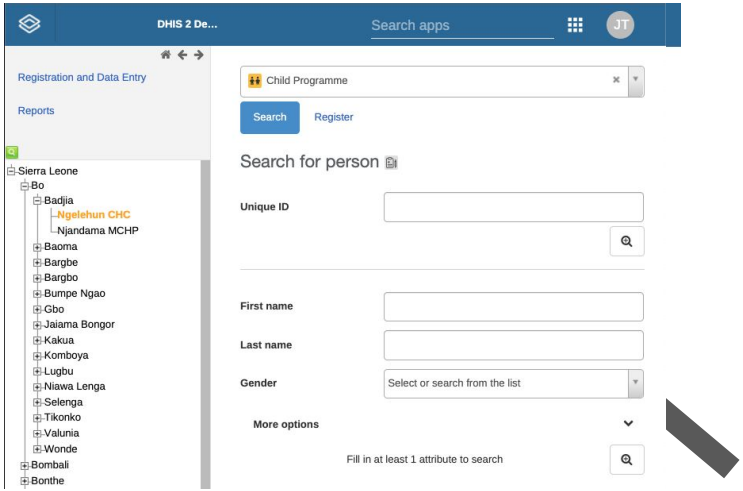
When IDs are generated by DHIS2, patterns using RANDOM are heavy on performance

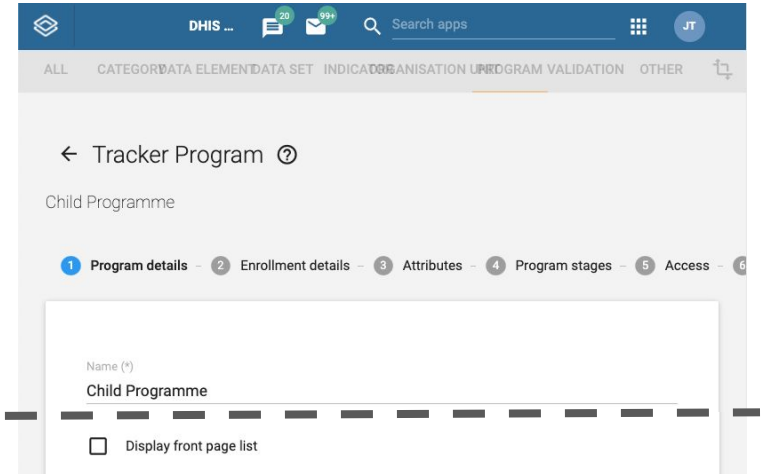


Migrate from RANDOM to SEQUENTIAL patterns

Tracker - Working lists

The standard working lists in tracker programs are generally heavy on performance





First aid: Turn off standard working lists

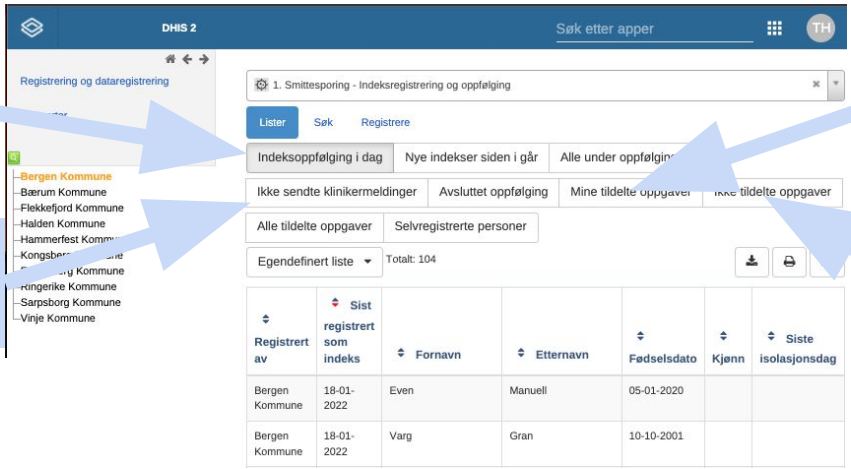
Drawback: All lists will also be disabled in Android

Tracker - Working lists

The standard working lists in tracker programs are generally heavy on performance

Indexes due for follow-up today

Notification not sent



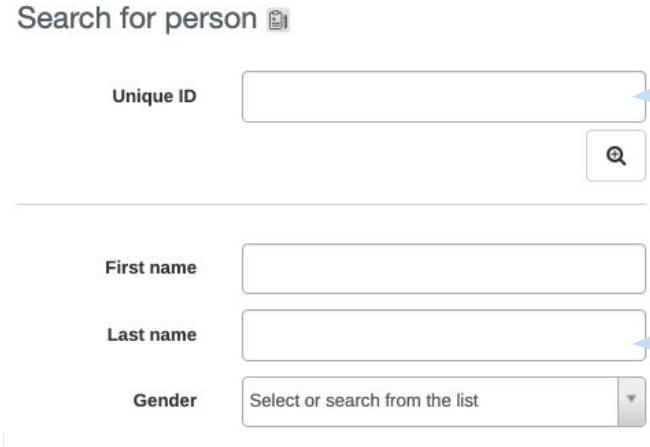
My assigned tasks

Unassigned tasks

Make targeted working lists containing relevant TEIs for specific tasks

Tracker - DB Indexes

Searches utilizing non-unique tracked entity attributes are heavy on performance



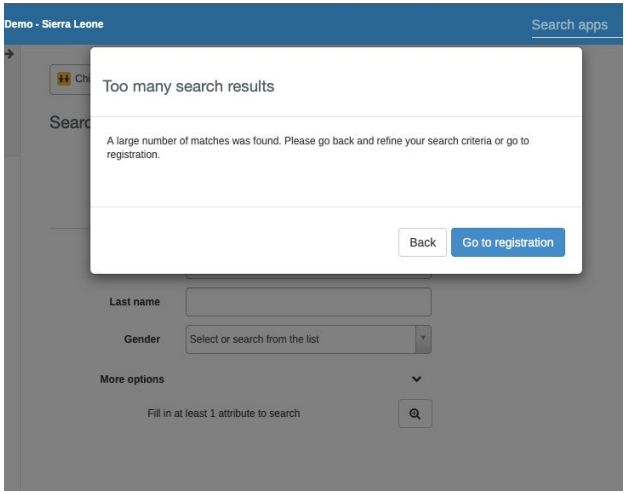
Unique attributes are fast

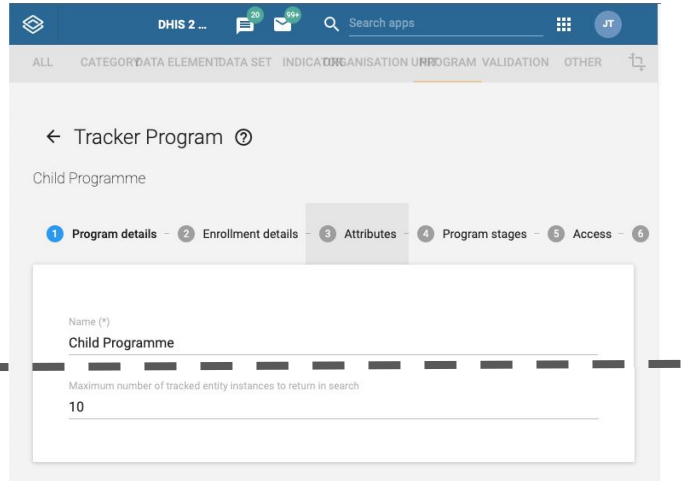
Non-unique attributes are heavy, and the btree indexes have a big positive impact on these searches

Add indexes to cover all searched attributes

Tracker - Search result limitation

Very broad searches will return a high number of records, and consume resources





Set a maximum number of records to return in search, for Program and Tracked Entity Type

Custom API queries

Custom scripts, custom apps and integration middleware has the power to make performance better - but also to make heavy and inefficient queries

skipPaging

pageCount

LIKE comparisons where EQ(uals) is possible

Check custom API calls for pitfalls - Monitor your system for unexpected heavy calls

Android considerations

DHIS2 Mobile Implementation at scale checklist	
User access configuration	
Auto-generated values pattern	
Android Settings Webapp: *Number of reserved values* *Automatic metadata sync period* *Automatic data sync period* *Data download settings*	
Management of Android App updates	
Mobile Devices Management	

Android - User Access

The application will try to download as much information as possible in case the device goes offline. To reduce the amount of data transferred:

- Minimize the number of Organisation Units, Programs and DataSets that users will have access to

Recommendations on how to create a user

Android - Auto generated values

Android will also download reserved values. The application will try to evaluate the amount of remaining values and retrieve more from the server whenever a synchronization happens.

- Be aware of the impact of using DATES and SEQUENTIAL patterns in offline settings
- For performance purposes avoid using RANDOM patterns

More information about this topic in the official documentation and in this post.

Android - Android Settings WebApp

The Android Settings WebApp helps you tailor your sync processes:

- Configure and adequate number of “reserved values”
- Adapt you periodic Data and Metadata sync processes
- Define your maximums download data based on the actual need
  - Tracker: number of TEIs, enrollment status, last update within, enrollment date within
  - Events: number of events, last update within
  - Data sets: number of periods to download
- Test you user specific sync simulation

More Info Here Android Webapp Configuration

Android - Mobile Device Management

Using a Mobile Device Management software can really help in big implementation. Main reasons are:

- Version Update control
- Locate and track lost devices
- Restrict or control the use of mobile data for specific applications

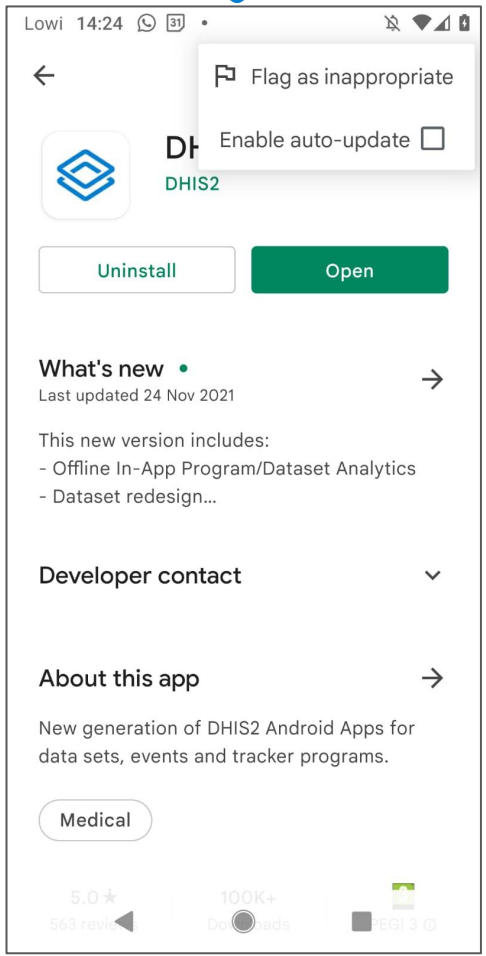
This MDM Guide covers several MDM that have been tested listing their main advantages and disadvantages.

Android - Management of Updates

Automatic Updates can be problematic and difficult to manage in big implementations.

- If you use Google Play for distribution, disable automatic updates.
- Have a testing plan based on your implementation needs and run it before every update

There is a chapter for testing in the Implementation guidelines.



Android solutions

DHIS2 Mobile Implementation at scale checklist	
User access configuration	✓
Auto-generated values pattern	✓
Android Settings Webapp: *Number of reserved values* *Automatic metadata sync period* *Automatic data sync period* *Data download settings*	✓
Management of Android App updates	✓
Mobile Devices Management	✓

Implementation Solutions

DHIS2 Metadata Package Resources

COVID-19 Vaccine Core Package (Aggregate/Dashboard)

Measure uptake, coverage & equitable distribution over time, by geography and across risk groups

- Daily vaccination report (based on tally sheet)
- Site-level stock reporting
- Target population data (denominators)

- Complete dashboards designed according to WHO NDVP monitoring recommendations: daily monitoring and overall coverage/vaccine uptake monitoring

Covid-19 Electronic Immunization Registry (EIR) Package (Tracker)

Ensure that individuals can be monitored for the full course of multi-dose regimen

- Electronic immunization registry
- Individual follow up, advanced drop out calculations
- Core WHO data set for digital certification requirements

- Daily/real-time dashboards for campaign monitoring, vaccines administered

Package resources & downloads: https://dhis2.org/metadata-package-downloads/#covac

EIR Tracker Package Updates for Performance

1. Tracker program configuration updated to optimize performance
    a. These can also be manually replicated in your country system if you have an adapted/customized version of the COVAC EIR tracker program.
2. Dashboard updates for more performant real time monitoring
    a. Tracker-based dashboards (e.g. those served by Program Indicators) have been removed
    b. Daily Monitoring Dashboard (based on the aggregate data model: indicators only) serves subset of real-time monitoring metrics based on tracker data
    c. Aggregate data set/data elements for transferring tracker data values to the aggregate model
3. Program indicator file
    a. You can use this file to update the configuration of package program indicators to incorporate the mapping to corresponding aggregate data elements/catOptionCombos (for tracker-to-aggregate)
    b. Some updates to configuration of program indicators (also specified in the guidance doc)

https://dhis2.org/metadata-package-downloads/#covac

*Be sure to check the updated Installation Guide and System Design Guide.

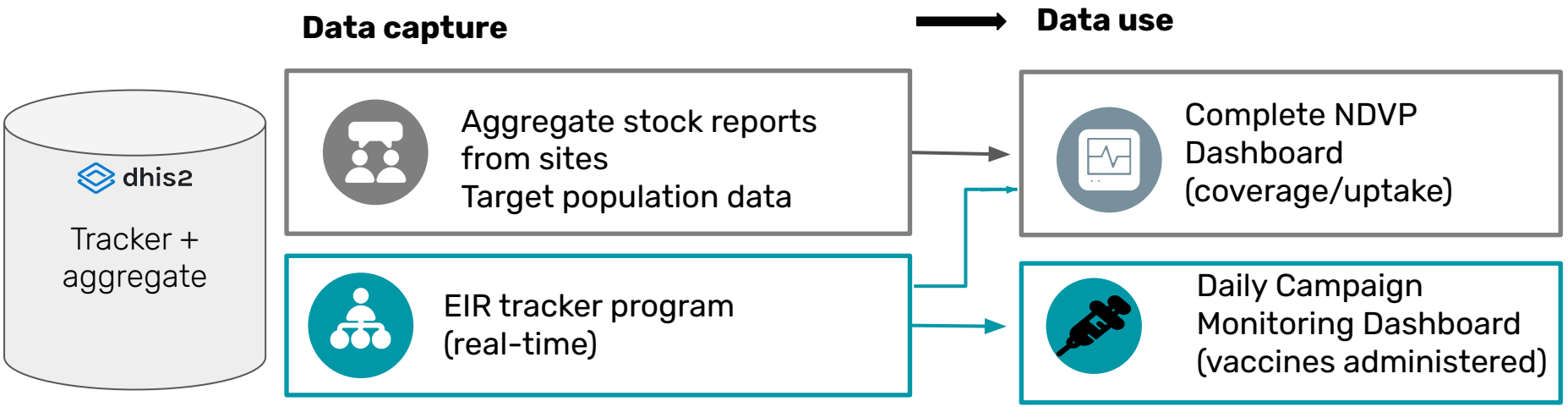
Implementation Strategies

Is your tracker implementation ready to use tracker data as the source for real-time monitoring?

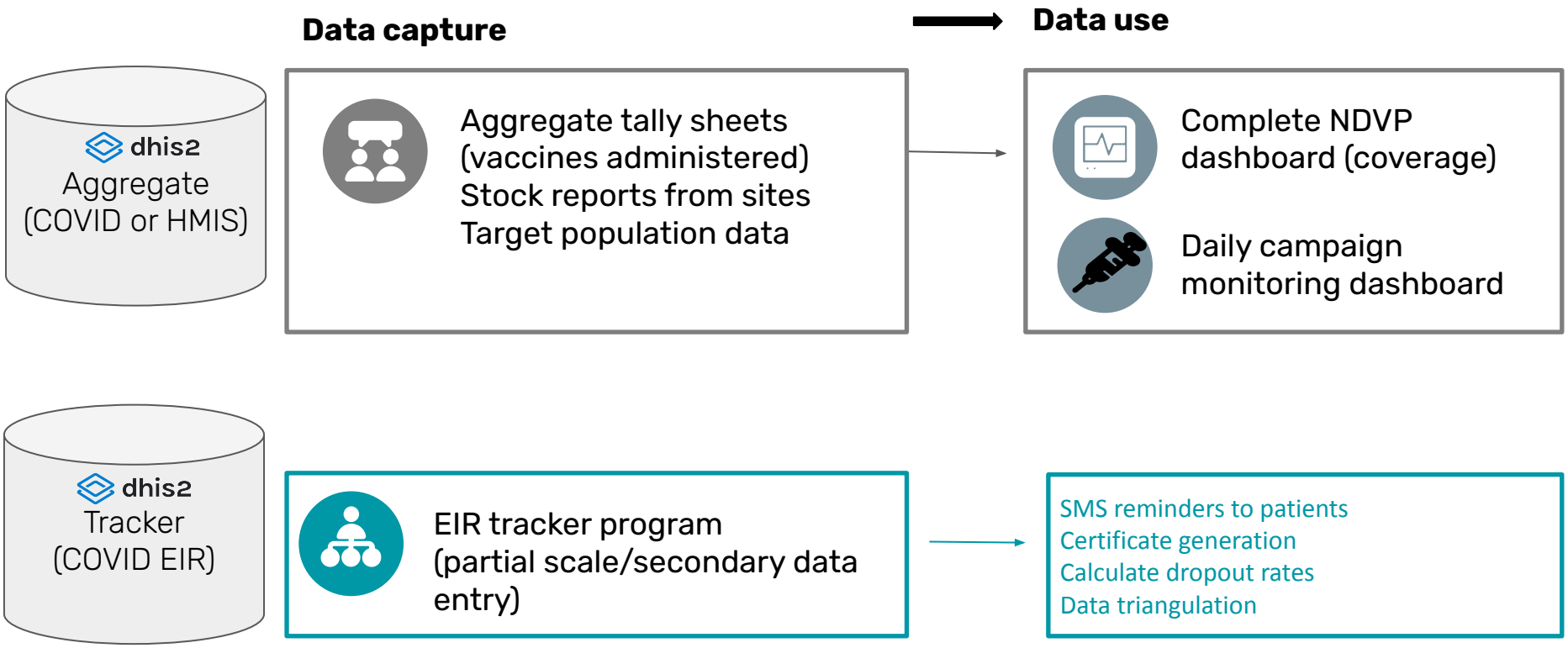
1. All vaccination sites are equipped with adequate number of usable devices, have stable internet connectivity to sync data at least once per day, and have sufficient number of trained data entry personnel?
2. The lag time for individual level data entry in DHIS2 is ~ less than one day?
    a. If there are large discrepancies between tally sheets and individual records, your implementation might not be ready to use tracker data as the source data for daily, real-time monitoring
3. Server hosting & monitoring function is staffed by a knowledgeable expert and analytics can successfully be run at least daily; there is minimal downtime disrupting data entry activities and data users can access updated dashboards at least daily?

If not, you may want to consider a hybrid approach to real-time monitoring using the aggregate data model!

Scenario 1: Tracker EIR at scale, real-time



Scenario 2: Parallel aggregate reporting



Tracker to aggregate

Rationale:

- Program Indicators are expensive, particularly with very large number of TEIs - counting rows is a very expensive database operation
- As discussed above, access to/use of PIs on dashboards may have to be restricted, but;
- Users still deserve to see some data

Approach:

- Map the most important program indicators and to aggregate data elements
- Periodically generate the aggregate data values from the program indicators
- Build simple dashboards based on the aggregate dataelements

Status of T2A in DHIS2

- Guidance on tracker to aggregate is available to support custom solutions: http://tinyurl.com/2p94893s
- Interoperability team is working on an automated tool which is coming soon
- Tests of performance and accuracy of T2A approach is ongoing - a few issues remain

Tool/script for automated T2A

Requirements

- The program indicators need to be in a program indicator group
- There must be a program indicator attribute which is used to indicate the code for the mapped data element
- The process is intimately linked with analytics - “normal” scheduled analytics must be replaced by script that runs analytics as part of T2A

The script

- Allows the big task to be broken down into smaller tasks (eg one API call per PI; or one API call per **n** orgUnits; or one API call per period)
- Tasks can be executed in parallel using a threadpool - being careful not to get carried away with too many parallel requests

We are still feeling our way to understand which parameters will give the right balance of efficiency and stability.

Support with COVAX performance

If you experience performance issues:

1) Read the tracker performance guidance documenting issues discussed here: https://docs.dhis2.org/en/implement/tracker-implementation/tracker-performance-at-scale.html
2) Go through the covax performance self-assessment/checklist: https://docs.google.com/spreadsheets/d/1LzQltZRtfCmSBgzXJ7zOpINpDdb4HTnDnUvx-ZaCeQs
3) Write to performance@dhis2.org so that the relevant people see the request and we can coordinate our support

Please do this *even* if you usually reach out to people on WhatsApp, Telegram etc.

Thanks.
Questions?