District Health Information Software 2

(DHIS 2)

Introduction

- Purpose
  - Demonstrate and provide information about DHIS 2 for project work
    - Demonstrate use of frameworks in a real-life application
- Contents
  - Domain model
  - Project structure & repository layout
  - Application design
  - Hibernate configuration
  - Testing
  - Development tips and conventions

DHIS 2

- Application for collection, validation, analysis, and presentation of aggregate / individual data

- Tailored to health information management
- Dynamic data model and flexible user interface
- Based on open source Java development frameworks
    - Globally distributed design practice

Domain model

- DataElement
  - Meta-data definition of captured enitities
  - Example: *BCG doses given for infant under 1 year*

DATAELEMENT	**Java-type**	**Db-attr**
id	int	unique, not-null
uuid	string	unique
name	string	unique, not-null
alternativeName	string	unique
shortName	string	unique, not-null
code	string	unique
description	string	-
active	boolean	-
type	string	not-null
aggregationOperator	string	not-null

Domain model

- Period
  - Time interval between start- and end-date
- PeriodType
  - Example: *Yearly, quarterly, monthly, weekly*

PERIOD	**Java-type**	**Db-attr**
id	int	unique, not-null
periodType	PeriodType	-
startDate	Date	-
endDate	Date	-

PERIODTYPE	**Java-type**	**Db-attr**
id	int	unique, not-null
name	name	not-null

Domain model

- OrganisationUnit
  - Example: *Ministry of Health, Province, District, Hospital, Ward*

ORGANISATIONUNIT	**Java-type**	**Db-attr**
id	int	unique, not-null
uuid	string	unique
name	string	unique, not-null
shortName	string	unique, not-null
organisationUnitcode	string	unique
openingDate	Date	-
closedDate	Date	-
active	boolean	-
comment	string	-

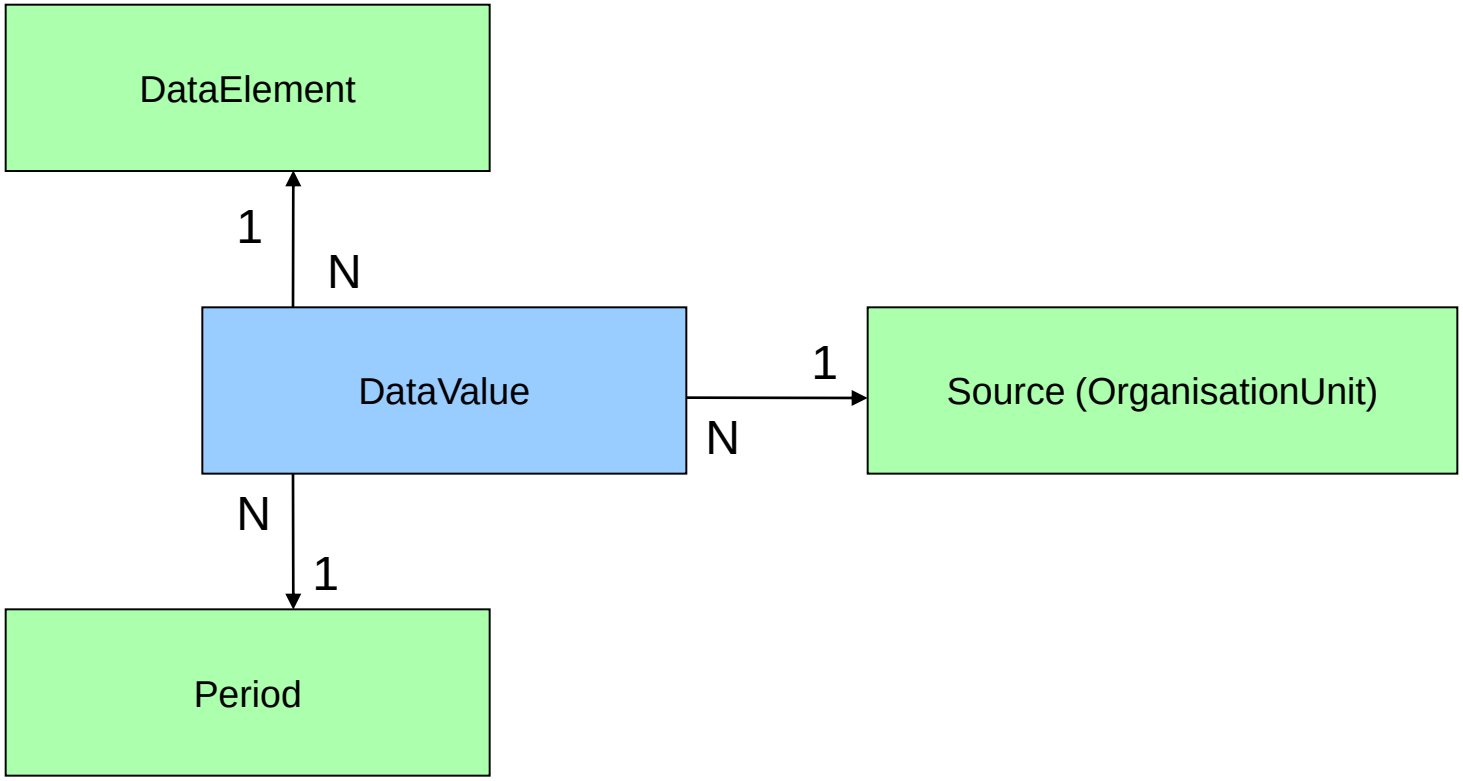
Domain model

- DataValue
  - Captured data for a combination of DataElement, Period, and Source

DATAVALUE	**Java-type**	**Db-attr**
dataElement	DataElement	-
period	Period	-
source	Source	-
value	String	-
storedBy	String	-
timeStamp	Date	-
comment	String	-

Domain model

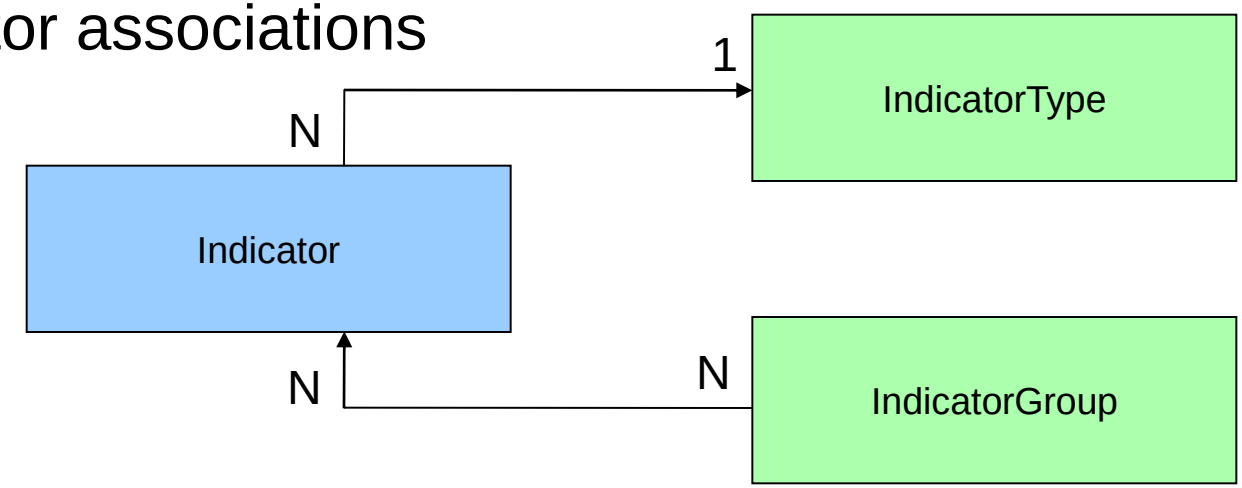
- DataValue associations



Domain model

- Indicator
  - Formula based on DataElements
  - Used to improve data analysis
  - Example: BCG under 1 year coverage = BCG doses under 1 year / Total population under 1 year
  - Multiplied with a factor (IndicatorType)

- Indicator associations



Domain model

INDICATOR	**Java-type**	**Db-attr**
id	int	unique, not-null
uuid	string	unique
name	string	unique, not-null
alternativeName	string	unique
shortName	string	unique, not-null
code	string	unique
description	string	-
indicatorType	IndicatorType	-
numerator	string	-
numeratorDescription	string	-
numeratorAggregationOperator	string	-
denominator	string	-
denominatorDescription	string	-
denominatorAggregationOperator	string	-

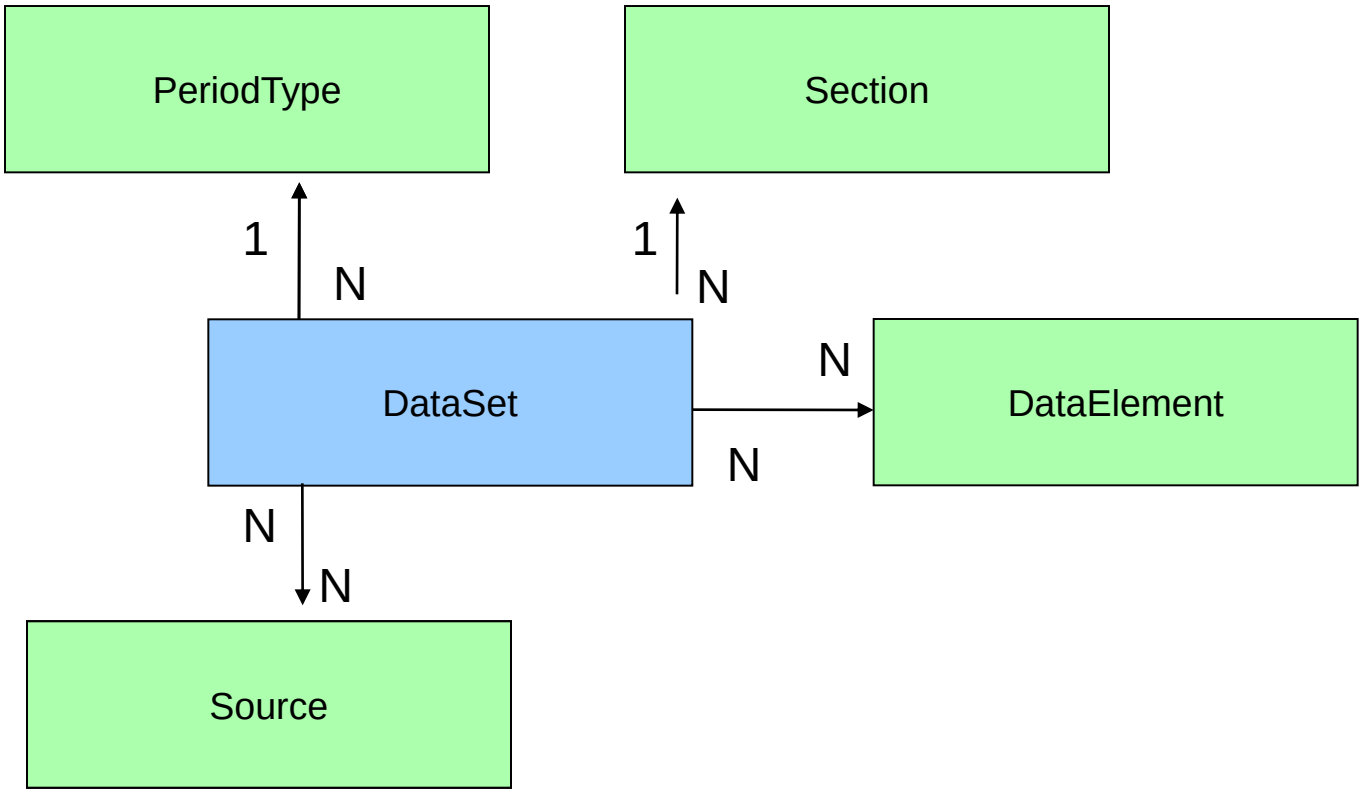
Domain model

- DataSet
  - Set of data elements
  - Used to organize data entry / input

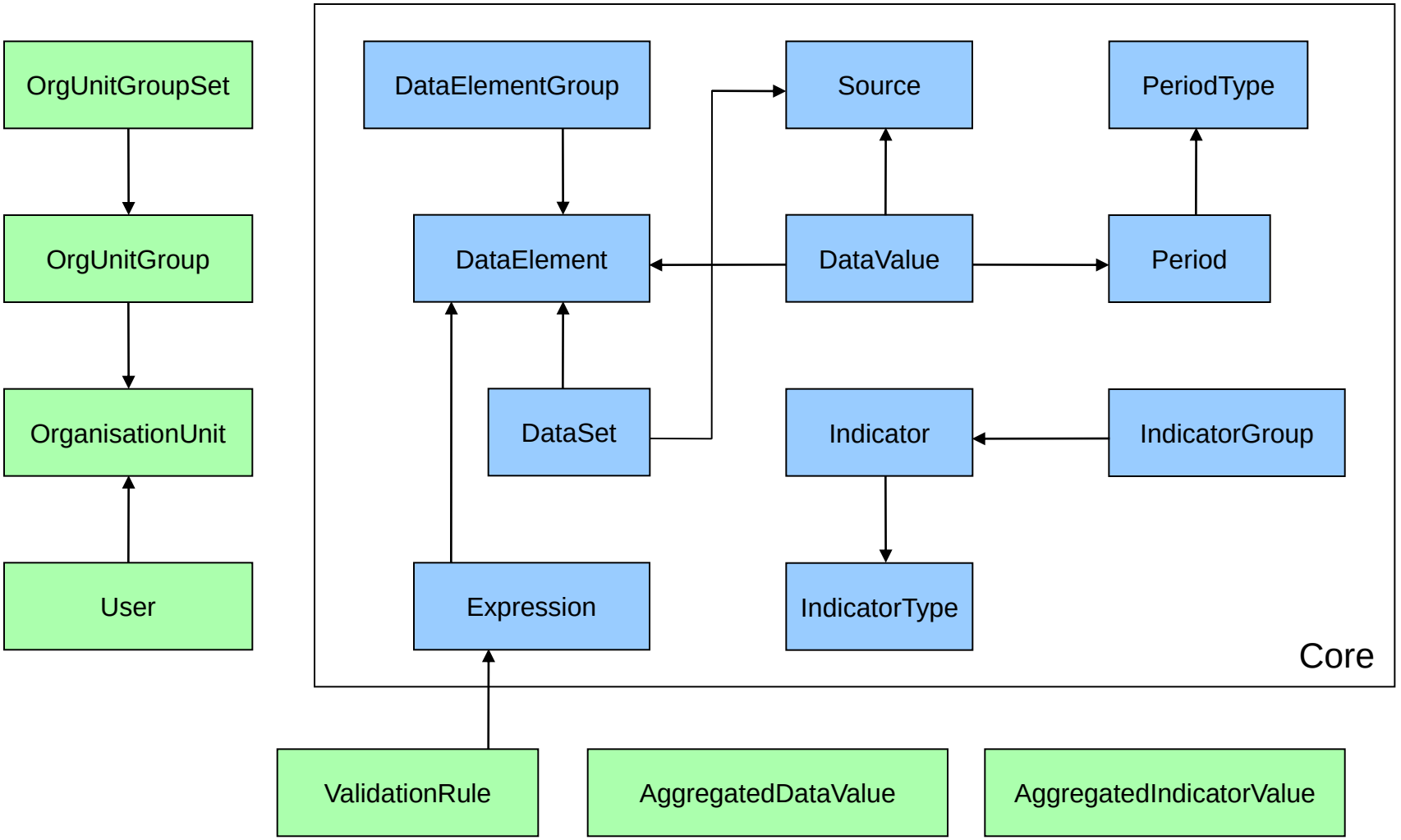
DATASET	**Java-type**	**Db-attr**
id	int	unique, not-null
name	string	unique, not-null
periodType	PeriodType	-
dataElements	Collection<DataElement>	-

Domain model

- DataSet associations



Domain model

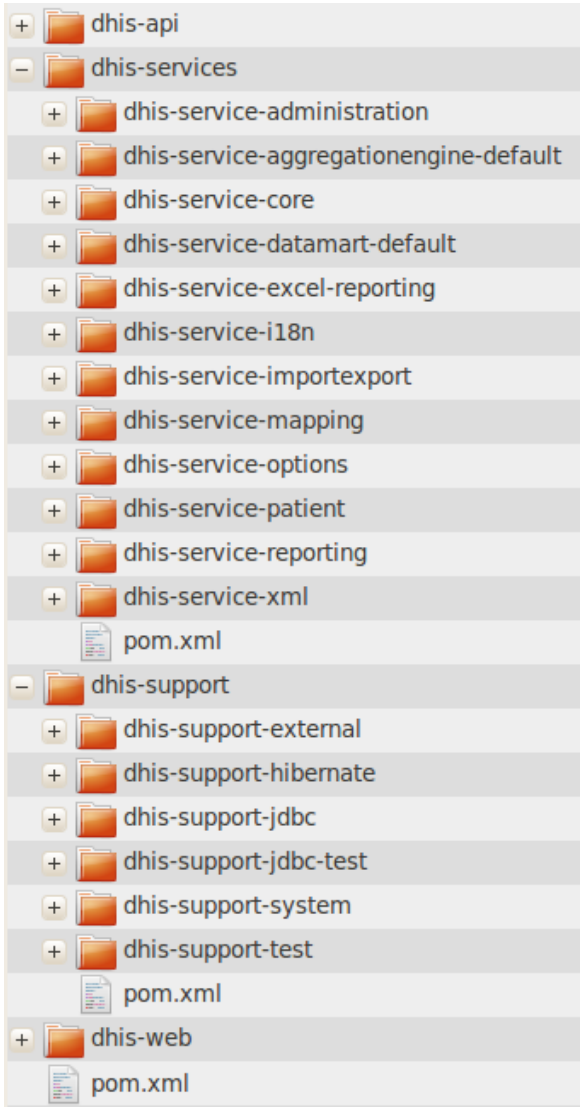


Project Structure Core

Service POM.
Root as parent,
contains project aggreg.
for service modules

Support POM.
Root as parent,
contains project aggreg.
for support modules

Root POM.
Contains dependency
management and
project aggregation.



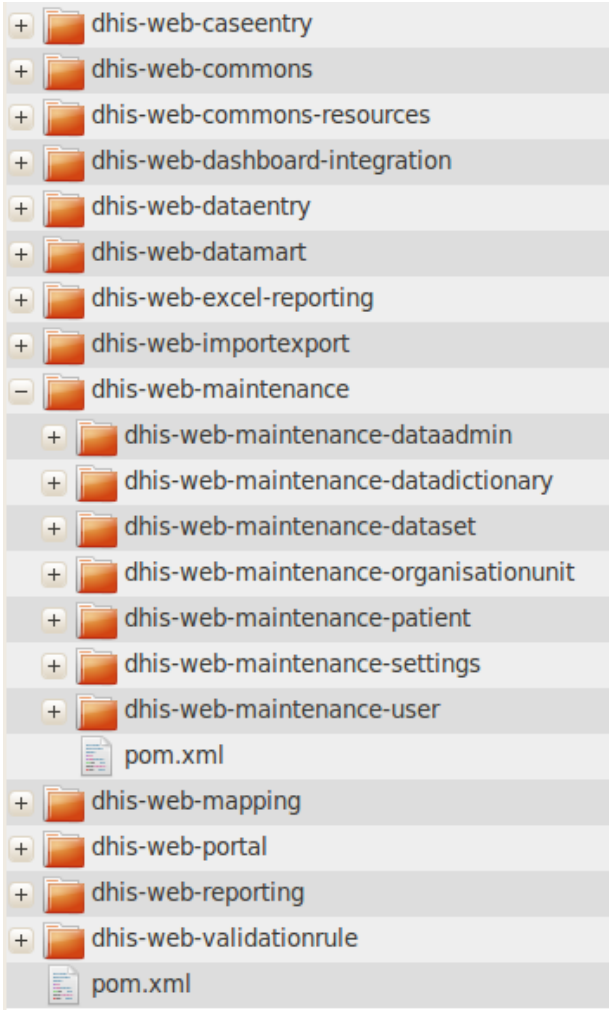
Building root POM
will build all projects
except dhis-web

All projects follow
the Maven standard
directory layout

Project Structure Web

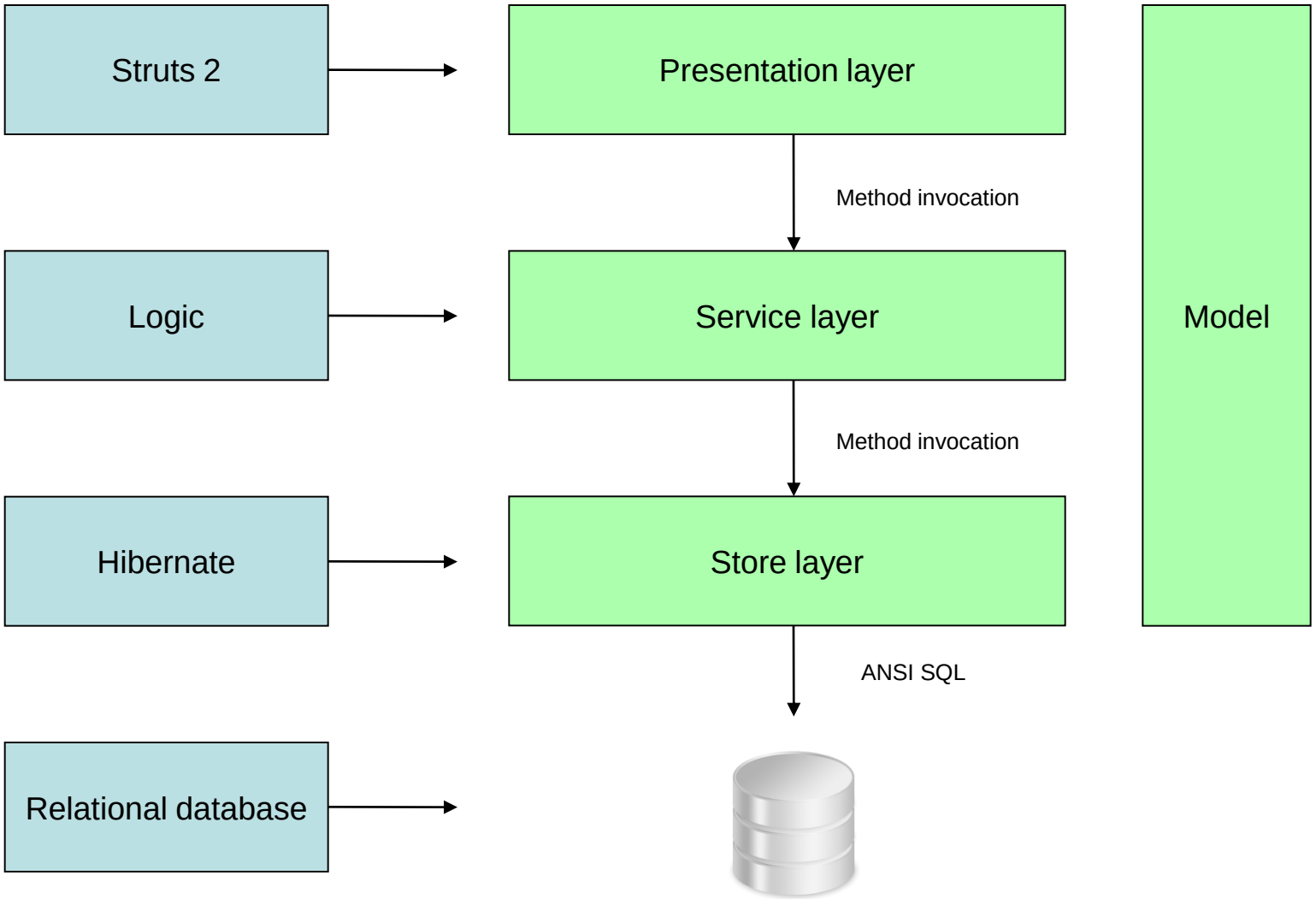
Web maintenance POM.
Web root as parent.
Contains project aggregation.

Web root POM.
Root as parent.
Contains dependency management and project aggregation.



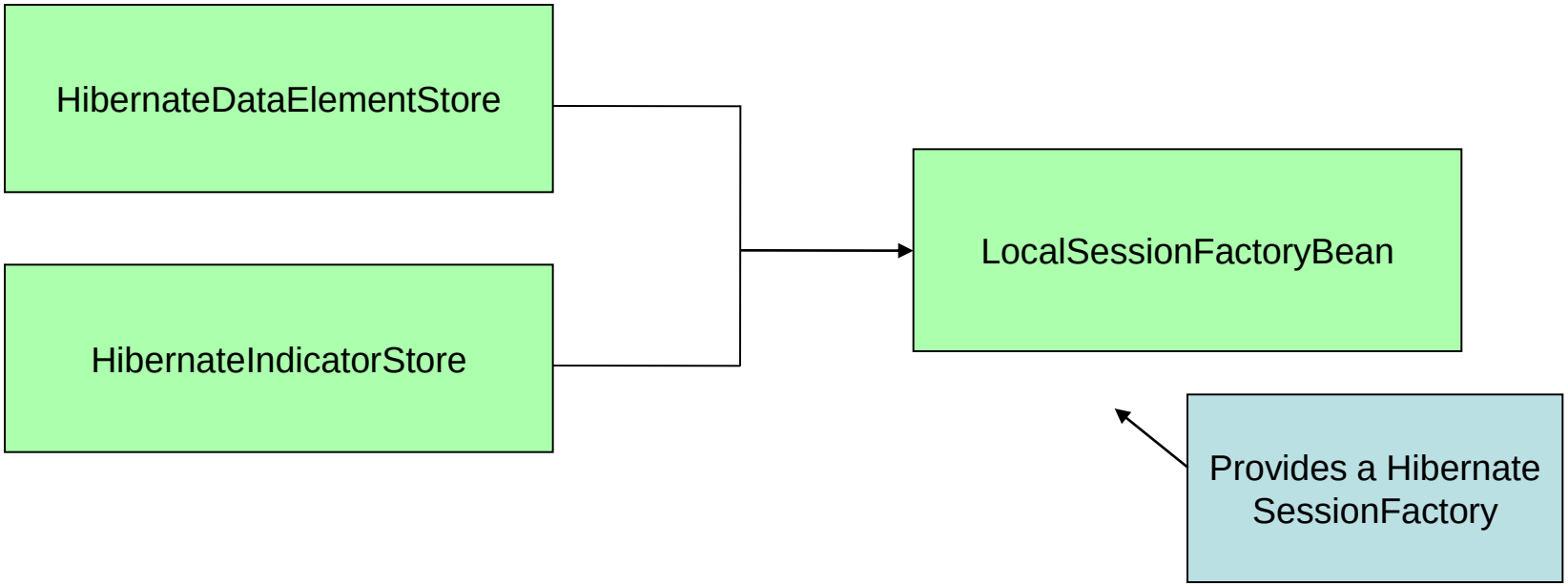
Building web root POM will build all web projects

Application design

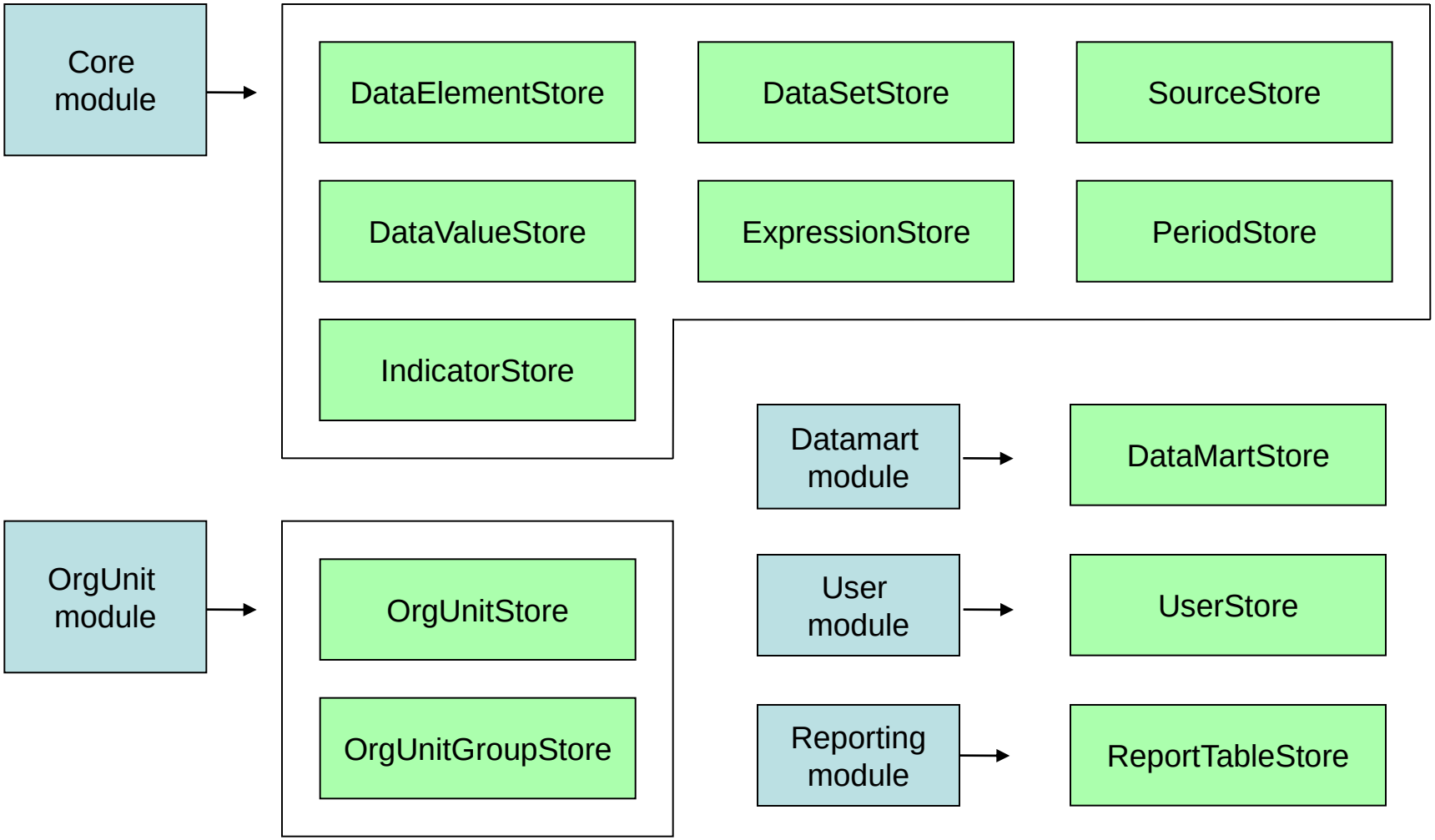


Store layer

- One *store* per domain object
- Performs CRUD operations
- Hibernate implementations of each store
- All stores have dependencies on the *LocalSessionFactoryBean (Spring)*

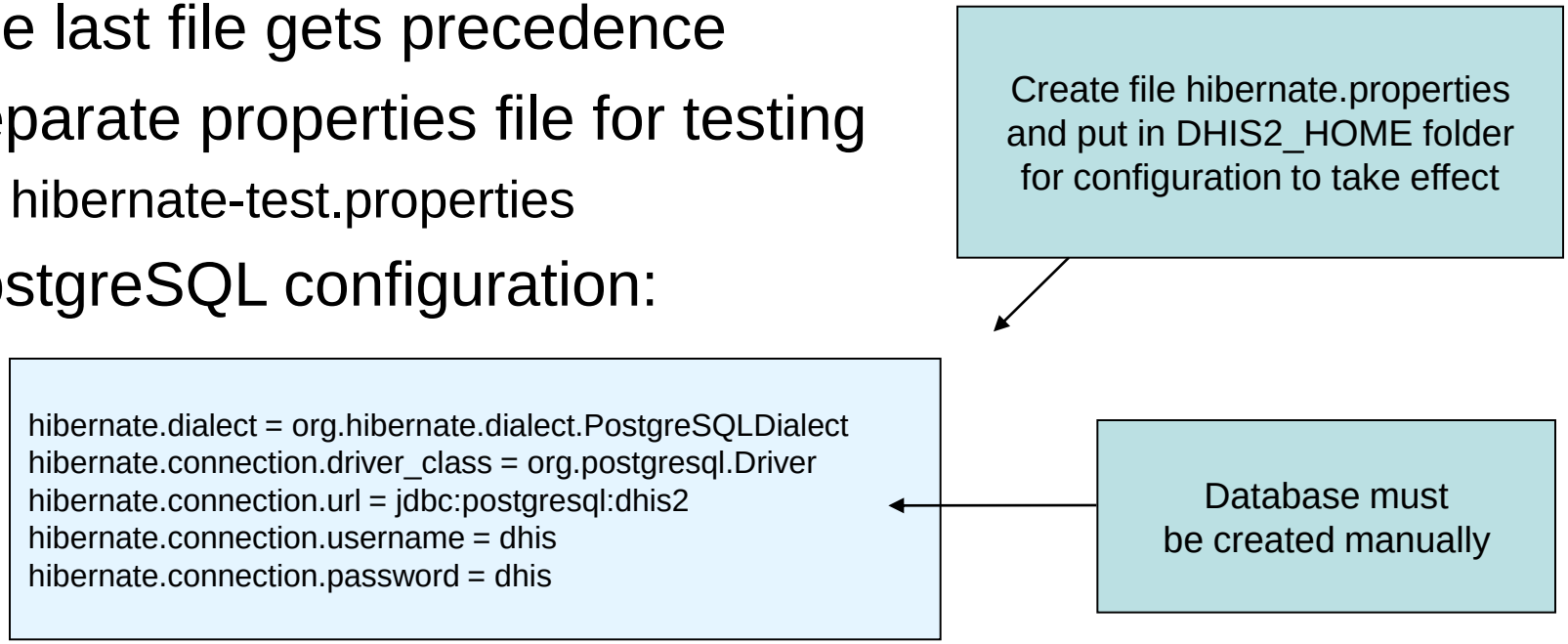


Store layer overview



Hibernate configuration

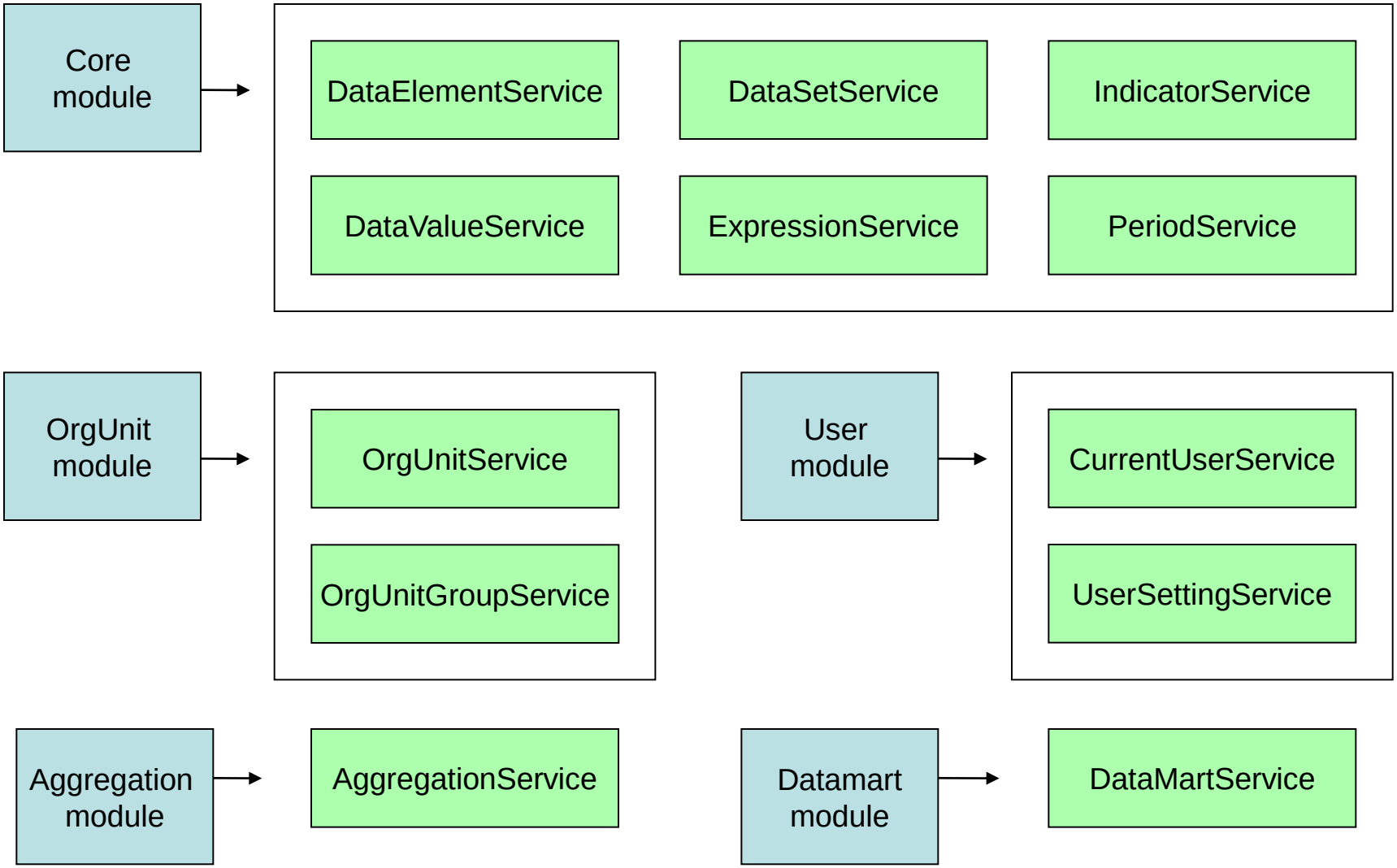
- ConfigurationProvider looks for *hibernate.properties:*
  - Inside dhis-support-hibernate (hibernate-default.properties)
  - On the classpath (src/main/resources) in all projects
  - In DHIS2_HOME directory (env variable)
- The last file gets precedence
- Separate properties file for testing
  - hibernate-test.properties
- PostgreSQL configuration:



Service layer

- Service functionality:
  - Administration
  - Settings
  - Datamart
  - Import and export
  - GIS / mapping
  - Reporting
  - User management
  - Validation

Service layer overview



Presentation layer

- *Struts 2* used as web framework
- Web modules can run independently or in the *web portal*
- The web portal assembles all web modules (WAR-files) into a complete application
- Common web functionality located in *dhis-web-commons*
- Common web resources located in *dhis-web-commons-resources* (css, javascript, templates)

Support projects

- DHIS Support External
  - Configuration manager
    - Write and read configuration objects
  - Location manager
    - Get file or stream for reading
    - Get file or stream for writing
    - Relative to the DHIS2_HOME environment variable
- DHIS Support System
  - Support functionality for processes, object deletion, startup routines and more
  - Util classes for codecs, dates, math, PDF, text and more

Installing DHIS 2

- Live package (all-in-one)
  - Embedded servlet container (Jetty)
  - Embedded database (H2)
  - Unzip and invoke exe file / sh script
- Manual installation
  - Install DBMS (PostgreSQL, MySQL)
  - Configuration setup
  - Install servlet container (Tomcat, Jetty)
  - Deploy DHIS 2 WAR file

Collaboration tools

- DHIS 2 on Launchpad
  - launchpad.net/dhis2
  - Check out with Bazaar with lp:dhis2
- Developer mailing list
  - Subscribe: launchpad.net/~dhis2-devs
  - Post: dhis2-devs@lists.launchpad.net
- DHIS 2 site
  - dhis2.org
  - Documentation, development, downloads, features, demo

Development conventions

- Follow the code style
  - Eclipse: http://dhis2.org/download/resources/dhis-code-style-eclipse.xml
  - Import in Eclipse: Properties – Java – Code style – Formatter – New
- Do not break the layered architecture
  - Stores for persistence
  - Services for business logic
  - Web modules for GUI
- Write informative commit messages

Development tips

- Create *working sets* in Eclipse to keep track of projects
- Use Maven Jetty plugin during web development
  - Invoke with: $ mvn jetty:run-war on WAR projects
- Speed up building with Maven:
  - Without tests: $ mvn install –DskipTests=true
  - One test among others: $ mvn test –Dtest=NameOfTestClass*