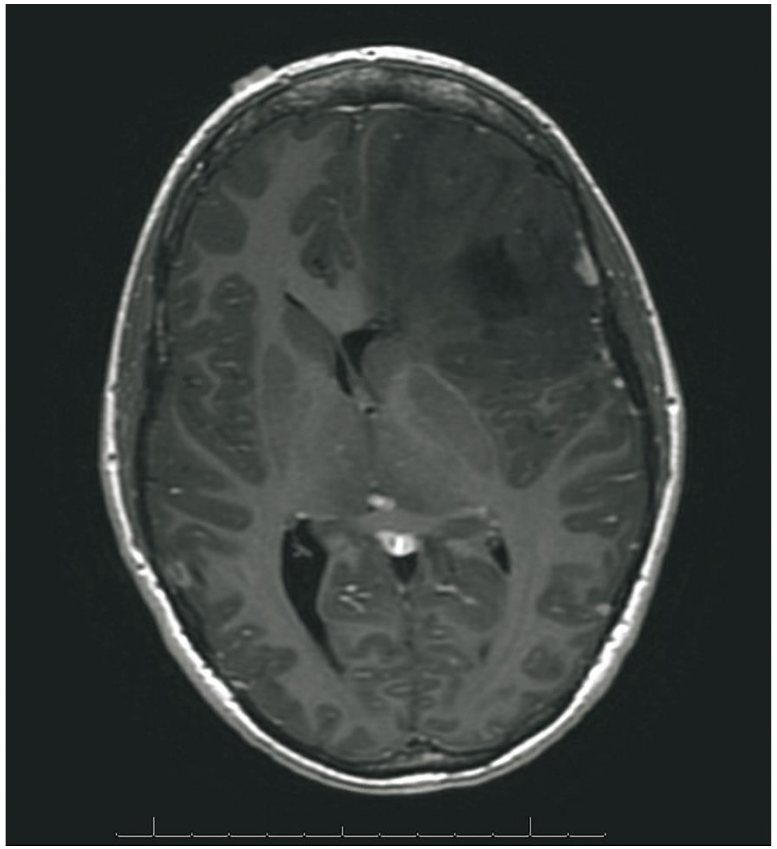
1. Securing Stata
2. Using characterisics for metadata

2023 UK Stata Conference

Bjarte Aagnes

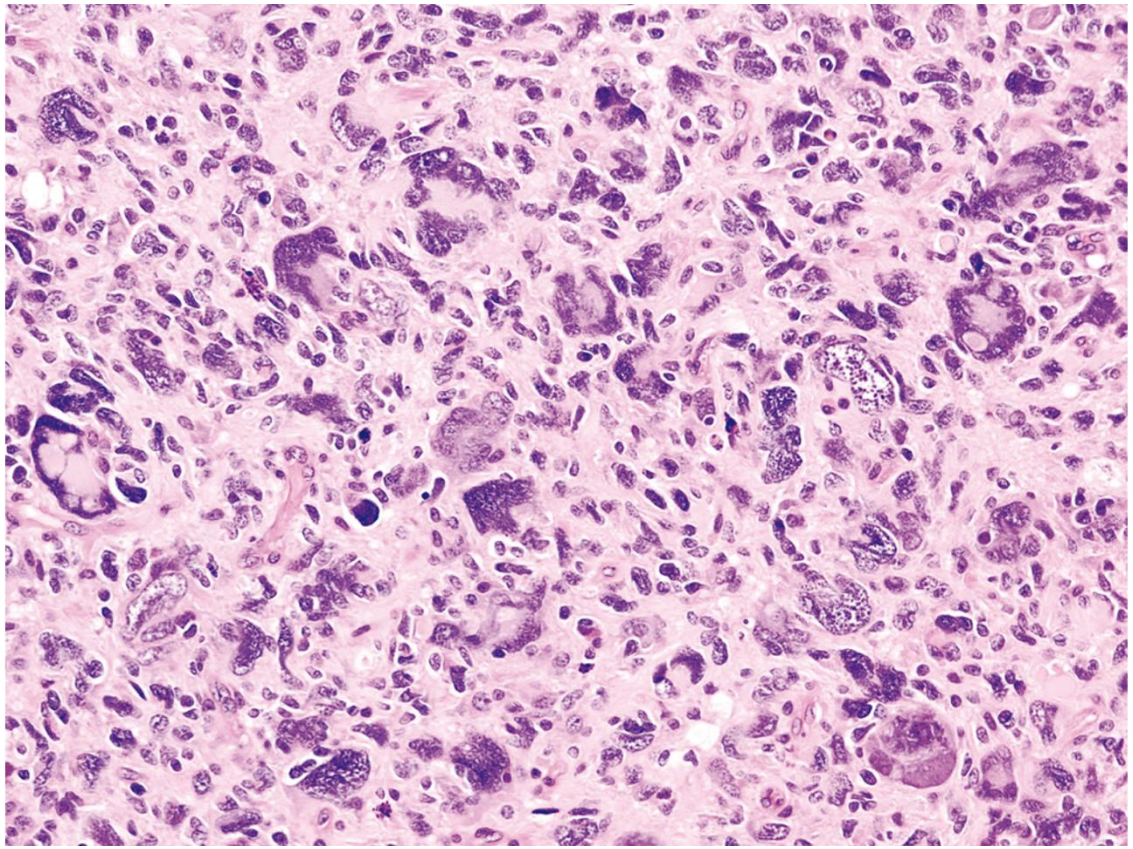
Cancer Registry of Norway

Cancer data start with cancer diagnosis



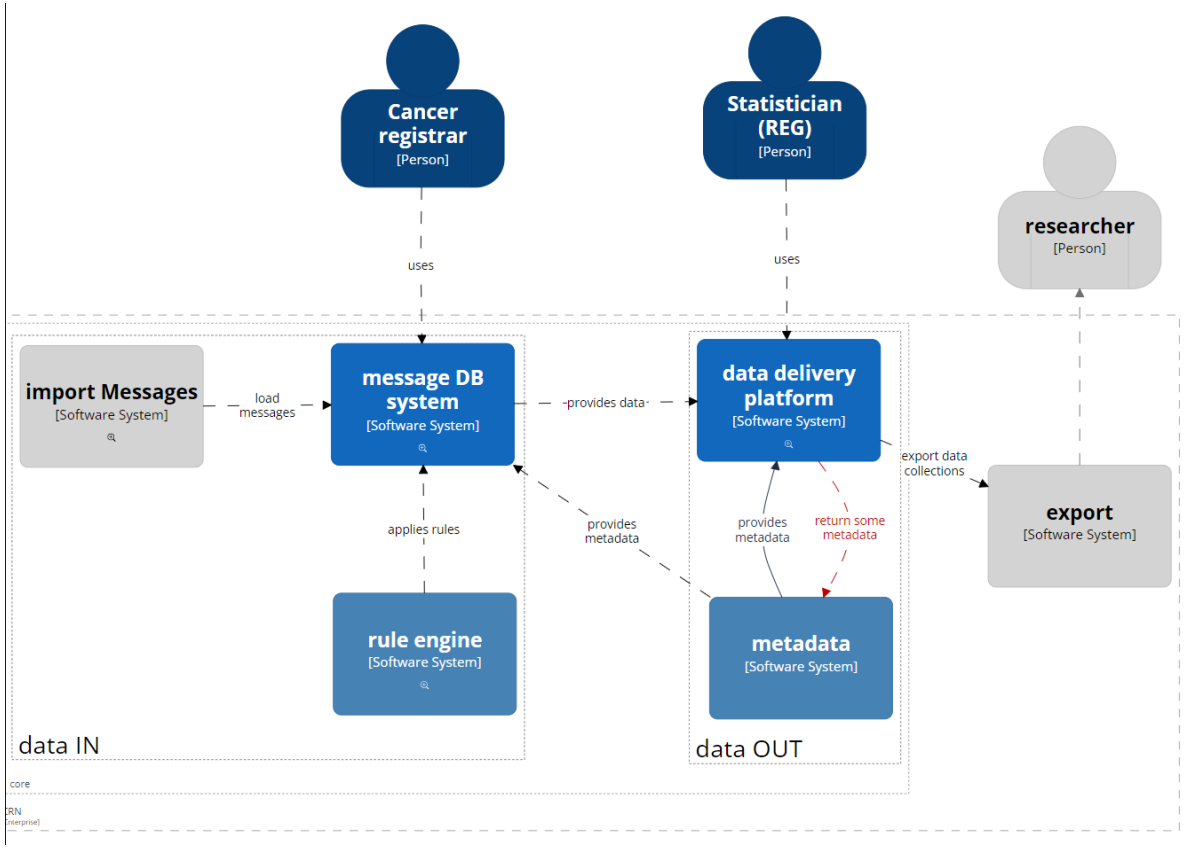
brain tumor

WHO classification of CNS tumors



morfology 94453 “Glioblastom, IDH-mutert”

Technical context (Cancer Registry of Norway)



System landscape

background/motivation

New data extraction and delivery project:

- legislation/regulations/corporate guidelines
- ultimatum: secure Stata use (or ...)

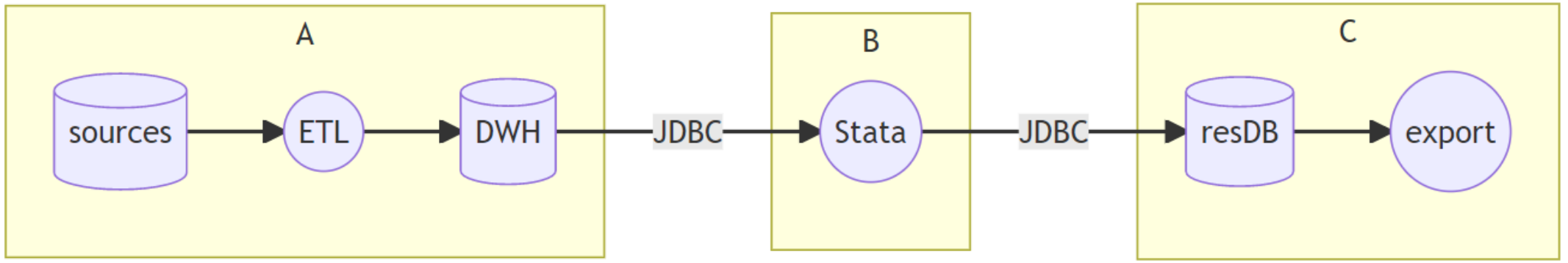
cancer patient data

- data integrity
- privacy of sensitive data (information on health issues)
- monitoring/logging: who did what when with data

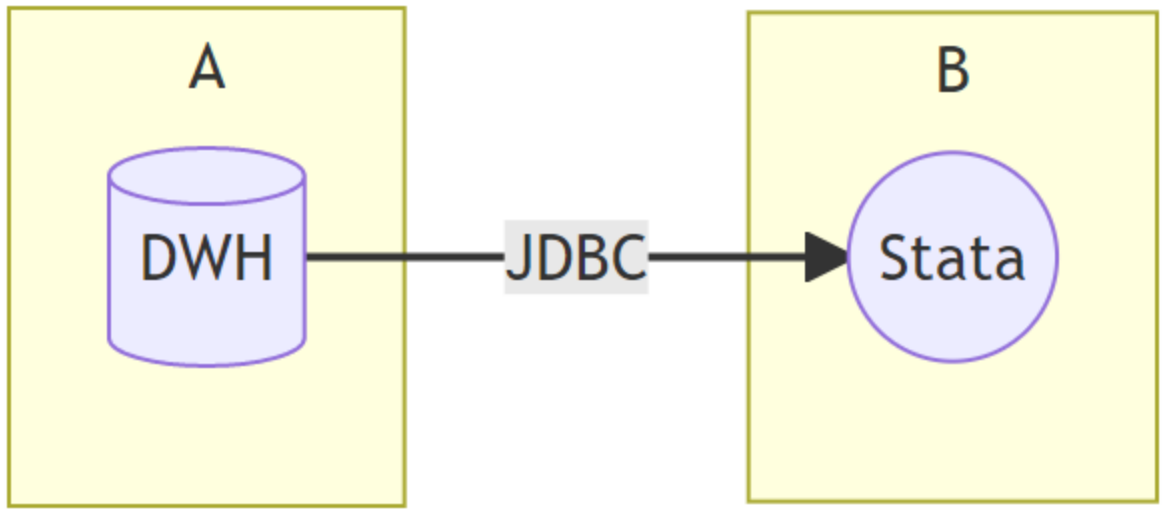
CHALLENGE: Stata is basically open:

- read/write data
- optional logging
- Stata jdbc add (clear text password)

CONTEXT: data flow



data flow

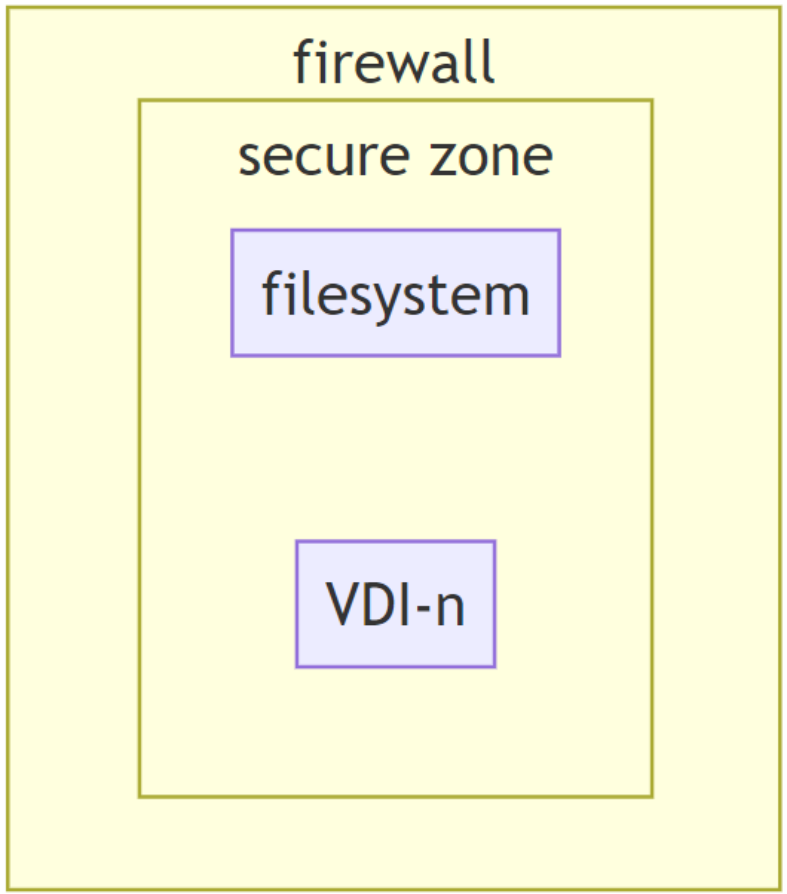


securing Stata

- define trusted/private network zone
- Java plugin for DB access and audit logging
- logging of Stata session

define trusted/private network zone

- firewall restrict data communication by rules
- private file server/file system (Stata network installation)
- personal clients (persistent VDI VMware Horizon Clients for Windows)

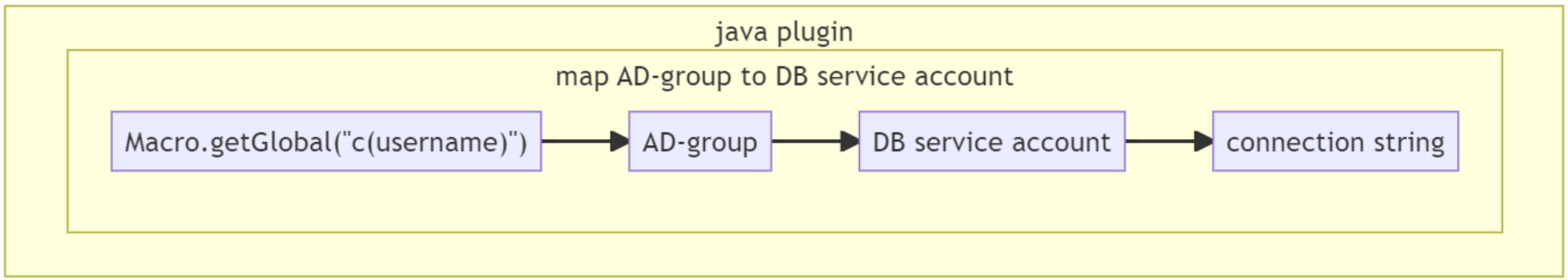


Java plugin for DB access via JDBC

- provides connection string
- hides JDBC connection password
- provides audit log of data load (who did what when)

Java plugin

- Reads *c(username)*. The user is *authenticated* when logging into the VDI
- Finds AD-group for user
- Maps AD-group to a *database service account*
- Decrypt and provide password for *database service account*



- The java plugin provides a connection string including a secret password stored (encrypted) in java resource file
- The user does not know the password
- The *DB service account* provide *autorization* to relevant data in DB
- Next verson will use Vault (HashiCorp) to provide short lived password (password rotation)

java code (elements) using Stata-Java API (https://www.stata.com/java/api17/)

```java
import com.stata.sfi.Macro;
import com.stata.sfi.SFIToolkit;

//Service stata commands to disable debugging and tracing
SFIToolkit.executeCommand("set trace off", false);
SFIToolkit.executeCommand("set debug off", false);

//Template connection string
String run_jdbc_add_dsn =
  "capture noi jdbc add " + jdbcConnectionValues.get("DSN") + ", " +
  "jar(\"" + jdbcConnectionValues.get("JAR") + "\") " +
  "driverclass(\"" + jdbcConnectionValues.get("CLASS") + "\") " +
  "url(\"" + jdbcConnectionValues.get("URL") + "\") " +
  "user(\"" + jdbcConnectionValues.get("USERNAME") + "\") " +
  "password(\"" + jdbcConnectionValues.get("PASSWORD") + "\")";

 //Execute the stata command
 int jdbc_add_dsn_result = SFIToolkit.executeCommand(run_jdbc_add_dsn, false);
```

What do the user do to fetch data from DB?

```
* typically define SQL select statment in filename.sql

KRG_KNUT_DWH_load, sqlfile("filename.sql")
```

```
prog define KRG_KNUT_DWH_load, ///
    nclass ///
    properties(KRG_KNUT_DWH)

findfile java_auditlog_plugin.jar // dependiencies (plugin)

version 17

syntax, sqlfile(string) [DIsplay]

confirm file "`sqlfile'"
scalar sql = fileread("`sqlfile'")
local sql = ///
    char(34) ///
    + trim(itrim(ustrregexra(scalar(sql),"\s|\x09|\x0a|\x0d"," "))) ///
    + char(34)

javacall krg.JavaAuditlogPlugin executeSql, jars(java_auditlog_plugin.jar)

if ("`display'" != "") {

        di _n _n  "`sqlfile'"
        di scalar(sql)
}

end
```

java code (elements) using Stata-Java API (https://www.stata.com/java/api17/)

```
//Get the sql from Stata
String sql = Macro.getLocal("sql");
String run_jdbc_load_sql =
 "capture noi jdbc load, exec(" + sql + ")" + " " + "clear";

//run Stata jdbc load
SFIToolkit.executeCommand(run_jdbc_load_sql, false);

//run standard Stata commands
SFIToolkit.executeCommand("noi compress", false);
SFIToolkit.executeCommand("noi describe", false);
```

KRG_KNUT_DWH_load

```
KRG_KNUT_DWH_load, sqlfile("filename.sql")
```

- Call the *executeSql* method, the only method exposed to user:
  - provide connection (jdbc add)
  - fetch data (jdbc load)
  - close connection (trick: jdbc add non-existing URL)

```
javacall krg.JavaAuditlogPlugin executeSql, jars(java_auditlog_plugin.jar)
```

Stata session log

- logging Stata session (started by sysprofile.do)
- monitoring Stata session log for illegal commands every x seconds (GoAnyWhere)
- moving Stata session log files for archiving (GoAnyWhere) for x months

What Stata properties made this possible?

- Stata jdbc command
- Stata-Python API (https://www.stata.com/python/api17/Data.html) for fast prototyping
- Stata-Java API (https://www.stata.com/java/api17/)

What would we add to Stata?

- JDBC close connection command
- run default commands at end of session (like sysprofile.do)

What did StataCorp fix?

- FIX update 04oct2022: jdbc load, when loading CLOB data, produced error when a NULL value was encountered.

What has been solved?

- restricting data reading/writing (plugin)
- logging of data reading/writing (plugin)
- avoiding clear text password (plugin)
- forced logging of Stata session including monitoring of content

Meta data

- Source meta data system
- In DWH tables Variable_Dim, Variable_Value_dim

- Daily creation of dta with only metadata
- For 3500 variables **26 variable metadata elements**
- For subset of variables **2 lists: value_code, value_code description**

- When fetching data meta data is automatically attached
- Add variable labels
- Add value labels if metadata «Integer» and lists of value_code, value_code description

Automatic labels from meta data

Variable name	Storage type	Display format	Value label	Variable label
PK_Message_Fact	int	%12.0g		primærnøkkel Message_Fact (krg_core_dwh)
sykdomstilfel~r	str7	%9s		Sykdomstilfellenummer
M_meldingstype	str1	%9s		Meldingstype
M_meldingsdato	double	%tc		Dato for undersøkelse
M_ds	byte	%211.0g	M_ds	Diagnosens sikkerhet
M_topografi	int	%115.0g	M_topografi	Topografi
M_topografiIc~3	str3	%9s		Topografi ICDO3

Automatic variable labels from meta data (EN)

Variable name	Storage type	Display format	Value label	Variable label
PK_Message_Fact	int	%12.0g		primary key Message_Fact (krg_core_dwh)
sykdomstilfel~r	str7	%9s		Case number
M_meldingstype	str1	%9s		Message type
M_meldingsdato	double	%tc		Date of examination
M_ds	byte	%211.0g		Certainty of diagnosis
M_topografi	int	%115.0g		Topography
M_topografiIc~3	str3	%9s		Topography ICDO3

Automatic value labels from meta data

M_topografi — Topografi

		Freq.	Percent	Valid	Cum.
Valid	341 Overlapp	3	30.00	30.00	30.00
	342 Midtlapp	2	20.00	20.00	50.00
	343 Underlapp	4	40.00	40.00	90.00
	349 Lunge uns	1	10.00	10.00	100.00
	Total	10	100.00	100.00	

Utility: get local macros for use in programming

```
s(category_name_en) : "Diagnostics"
   s(category_name) : "Diagnostikk"
         s(version) : "5"
       s(data_size) : "4"
  s(data_type_name) : "Integer"
         s(name_en) : "Topography"
  s(description_en) : "Origin of primary tumor. Mainly coded by ICD-O-2."
     s(description) : "Primærtumors utgangspunkt. Hovedsakelig kodet etter ICD-O-2."
            s(name) : "Topografi"
        s(tech_name) : "topografi"
              s(id) : "9"
 s(PK_Variable_Dim) : "8"
```

Thanks