



B O N C O D E

Code analysis Validatie

Customer: GVB

Project: SPOT

6 april 2021

J. Meetsma

j.meetsma@boncode.nl

Gaining objective insights

TO CREATE BALANCE

- Objective insights are vital to long-term success
- Without all relevant facts on the table, it's hard to make meaningful choices
- Create balance between short-term and long-term priorities in an objective way



Topics

Context

Meetmethode / statische code analyse

Bevindingen high level

Illustraties bevindingen

Overige bevindingen

Deep dive



Onderzoeksvragen

- “Onderwerp de SPOT-applicatie aan een Code Review en voorzie GVB van bevindingen, aanbevelingen en risico’s aangaande de codekwaliteit en doorgevoerde software architectuur.”
- Afgeleide vragen voor dit onderzoek met het oog op de komende aanbesteding:
 - Wat is de *onderhoudbaarheid** van de huidige codebase?
 - Is de codebase (inclusief documentatie) kwalitatief goed genoeg om het beheer en onderhoud (inclusief doorontwikkeling) van SPOT door de huidige leverancier of door een andere partij voor een periode van tenminste 5 jaar te kunnen laten uitvoeren?
 - Focus op *analyseerbaarheid* en *wijzigbaarheid*.
 - Welke verbeteringen dienen eventueel te worden doorgevoerd om dit verantwoord te laten doen?

**) De kwaliteitseigenschap ‘Onderhoudbaarheid’ van de ISO 25010 norm luidt: “De mate waarin een product of systeem effectief en efficiënt gewijzigd kan worden door de aangewezen beheerders.”*



Scope & deliverables

In scope:

- Java broncode: geautomatiseerde analyse en handmatige inspectie.
 - Test code volgens Maven conventie apart doorgemeten.
 - Test (utility) code in productiecode als test code beschouwd.
 - Code aangeduid met “tools” buiten scope gelaten.
 - Code in entity packages beschouwd als gegenereerd en buiten scope gelaten.
- Inventarisatie overige technologieën.
- Aangeleverde documentatie.

Deliverables:

- Een eindpresentatie in PowerPoint met daarin opgenomen de doelstelling van de opdracht, de onderzoeksvragen, bevindingen, aanbevelingen en mogelijke risico's.
- Overzicht architectuur c.q. topologie van het systeem.
- Dashboard op basis waarvan ontwikkelaars verbeteringen door kunnen voeren.



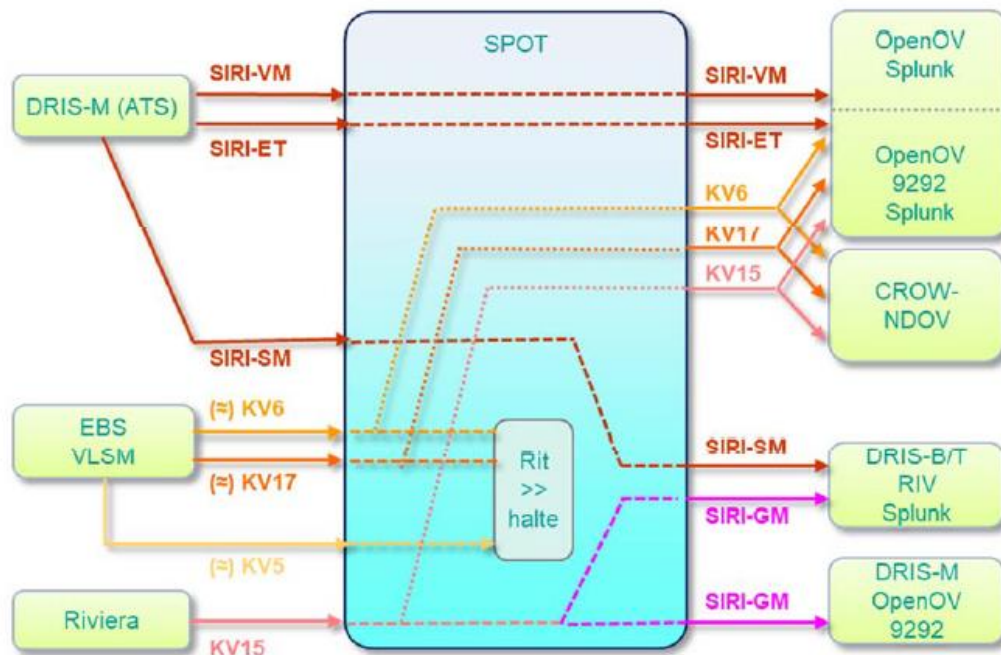
Sources

Source	Description	Date	Involved
SPOT IRS SIRI-client_services 1.4.3.docx		27-2-2021	Jetro Veenstra
SPOT SDD V2.2.1.docx			” ”
SPOT-DBDD_v24.docx			” ”
spot.zip	Main sources	23-3-2021	Erwin van Dijk
spotcommon.zip	Additional (re)sources		” ”
HLOB SPOT v1.8 definitief.pdf			Erwin van Dijk
SDD SPOT V2.4 4.pdf			” ”
SPOT IRS SIRI-client_services 1.4.3.pdf			” ”
SPOT SIRI use cases_v11.pdf			” ”
SPOT-DBDD_v25.pdf			” ”
SPOT-SVD_4.0.0.pdf			” ”



Achtergrond

- “SPOT is een software pakket ontwikkeld voor het GVB, waarmee alle datastromen die het GVB naar afnemers verstuurt, via één systeem verstuurd en gemonitord worden. Dit systeem is opgedeeld in verschillende deel-systemen, die elk hun eigen datastromen verwerken.”

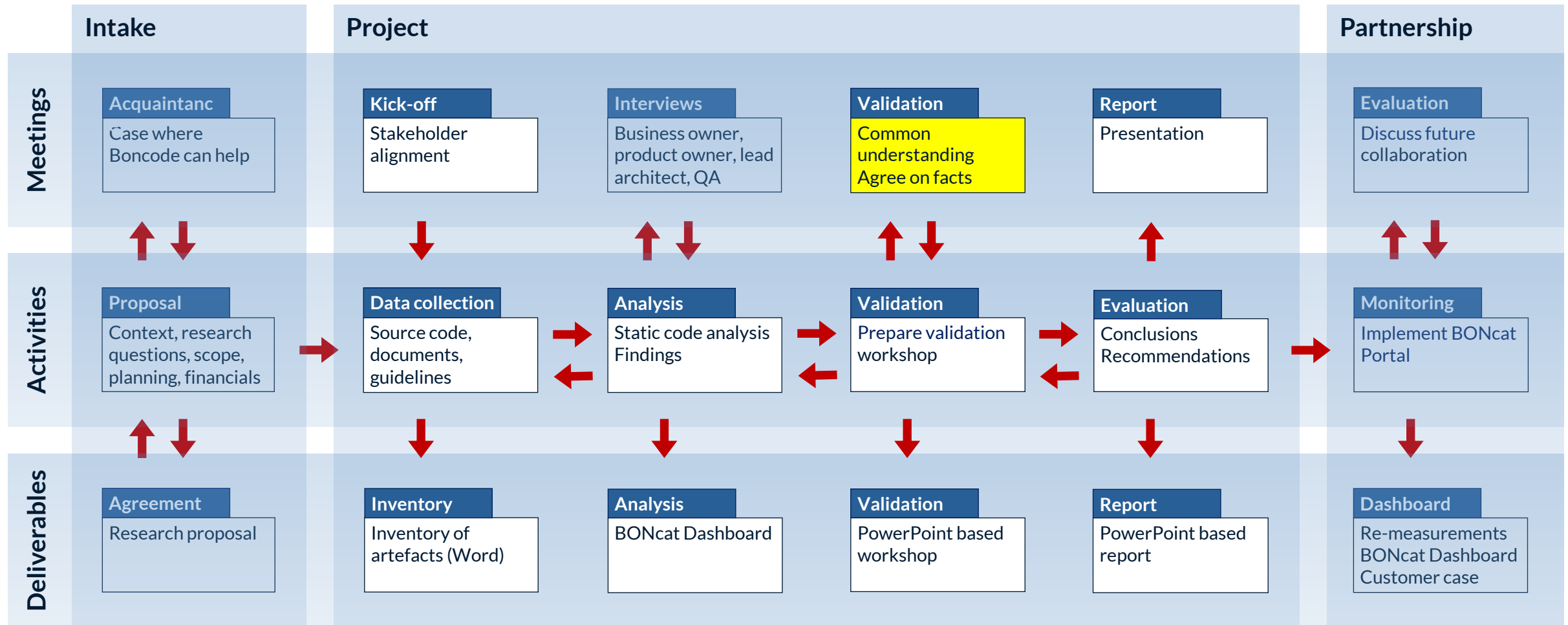


Taken SPOT:

- Transformeren,
- verrijken,
- bufferen,
- en routeren van berichten m.b.t. reizigersinformatie.
- Faciliteren van monitoring en besturing van berichtenverkeer / datastromen middels GUI.



Current stage research process



Doelstellingen gezamenlijke processtappen

Kick-off

- Alle betrokkenen informeren over de aanpak voor het onderzoek, de input die van eenieder verwacht wordt en een korte introductie in het vakgebied van software metrieken.

Interview(s)

- Inzage krijgen in projectverloop, software development processen (en zo mogelijk businessplannen en strategie).

Validatiesessie

- Gedetailleerde uitleg statische code analyse.
- Draagvlak / gezamenlijk beeld realiseren bij de betrokkenen voor de uitkomsten van het onderzoek.
- Presenteren en valideren bevindingen, onder andere aan de hand van voorbeelden.
- Vaststellen eventuele *false positives*.

Rapportage

- Tussentijdse impressie van stand van zaken.
- Eindrapportage met beantwoording onderzoeksvragen, belangrijkste bevindingen, conclusies en aanbevelingen.



Analysis method

Static code analysis



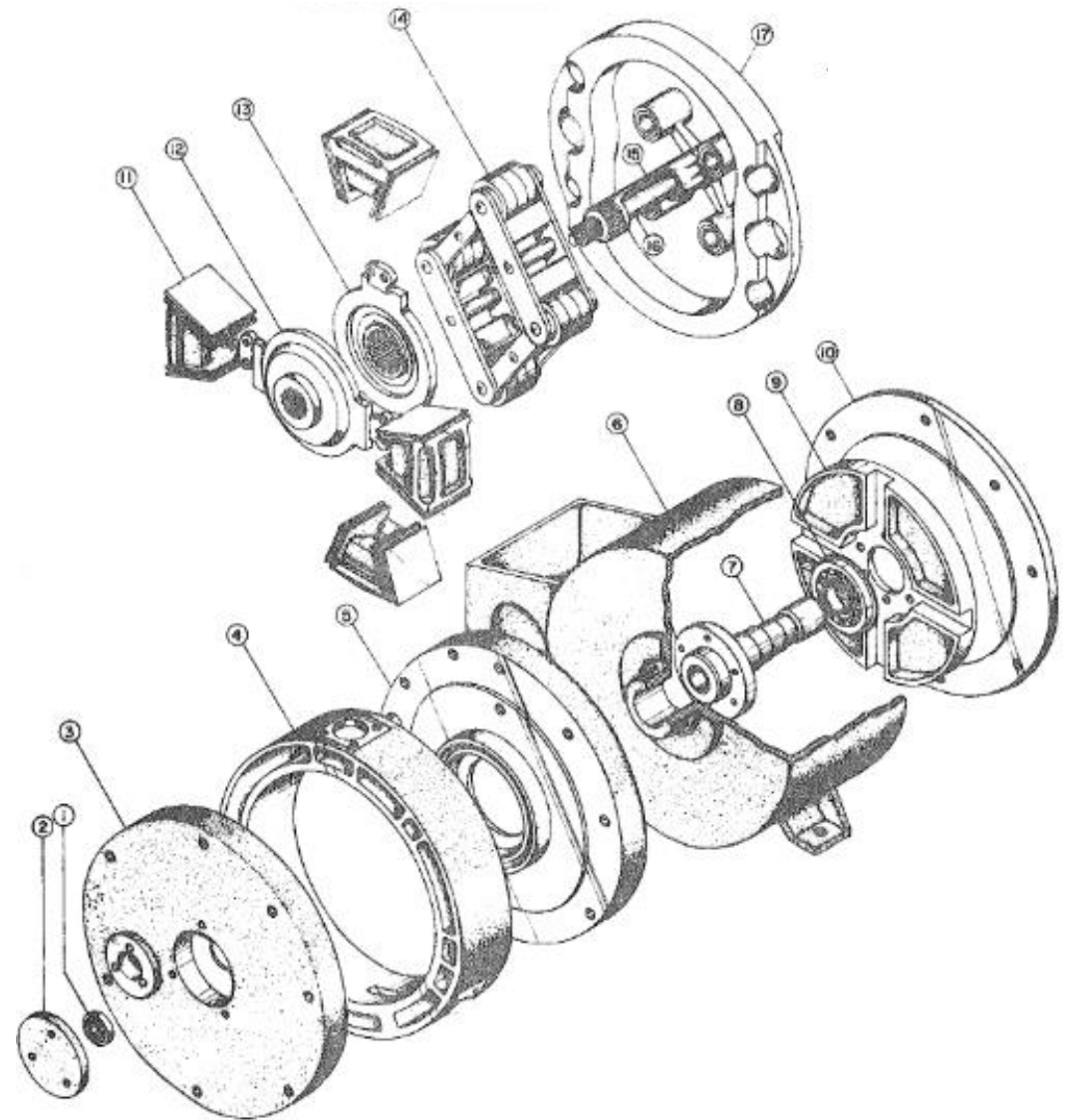
What is code?

In essence:

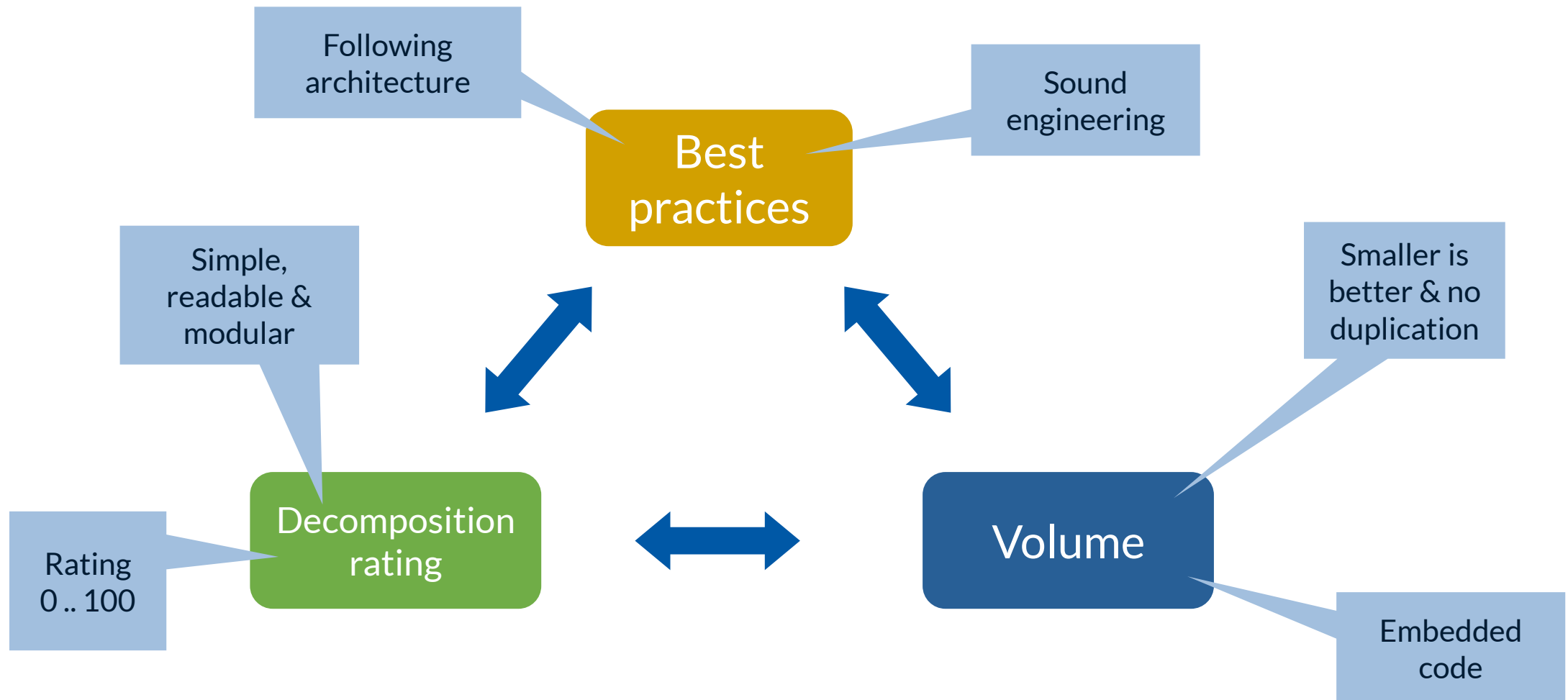
- A set of instructions for a machine (compiled)
- Divided into building blocks

But also:

- A description of the application's building blocks
- A description of how an application works
- A manual on how to maintain it



3 types of metrics used for analysis



Static code analysis primarily focussed on *decomposition*

- Main / basic train of thought: developers should continually split up the problem domain into coarse-grained and fine-grained modules ...
- ... that can be understood, tested and further developed in relative isolation
- When this practice is upheld, maintenance and development effort:
 - can be divided over multiple teams
 - will produce ever-increasing business value
- When this practice is abandoned, technical debt will increase and productivity will decrease

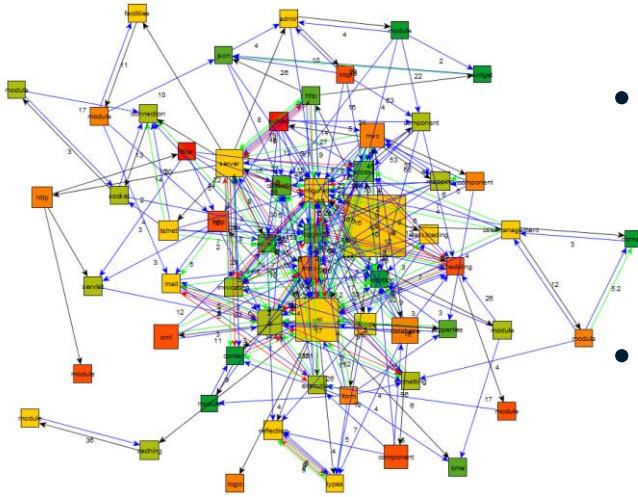


Decomposition

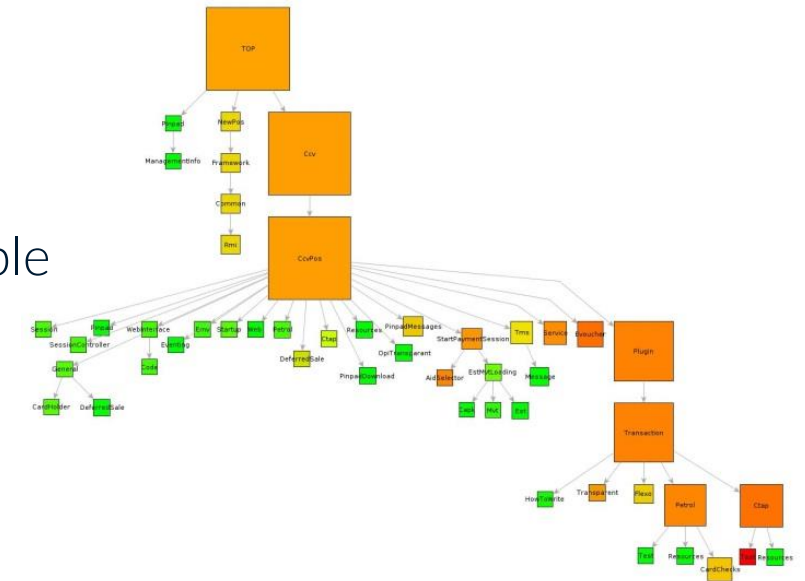
Metrics & findings



Background: Decomposition rating



- Established using 5 metrics regarding size, complexity and dependencies of Actions, Entities and modules
- Indicates how well developers have succeeded in:
 - dividing the business domain and technical domain into compact, testable units ...
 - ... arranged in modules with clear dependencies
- Is therefore an indication of attention and craftsmanship



Code Decomposition Metrics

Simplicity

- Functional decomposition
 - Method size
 - Method complexity

Method
metrics

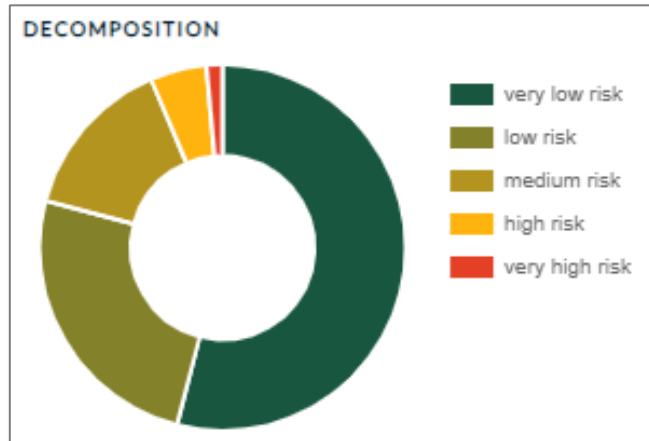
Modularity

- Interface stability
 - # Method parameters
- Module decomposition
 - Dependency span
 - Dependency volume

Module metrics



SPOT Decomposition rating is 68 /100



DECOMPOSITION	
Weighted Decomposition Rating	67.97
Function Parameter Rating	92.49
Function Size Rating	46.37
Function Complexity Rating	94.98
Dependency Span Rating	65.52
Dependency Volume Rating	54.78

- Overall rating kwalificeert onderhoudbaarheid als *adequate*
- Subratings geven indicatie dat
 1. code zich bevindt in relatief:
 - grote classes
 - en lange methodes
 2. code niet complex is c.q. uitgeschreven in reeksen van op zichzelf eenvoudige statements
- 94 % van de code is aangemerkt als laag tot medium risicovol
 - 1,4 % als zeer hoog risicovol

Expressiveness needs work

Low-level Code

```
private boolean collidesWithBlockSouth (  
    double spriteX, double spriteY,  
    double calculatedX, double calculatedY){  
    int y = (int)(calculatedY+28) / 100;  
    int x = (int)(calculatedX+28) / 100;  
    int x2 = (int)(calculatedX) / 100;  
  
    if( level[y][x] == X || level[y][x2] == X){  
        return spriteY < (y * 100);  
    }  
    return false;  
}
```

Expressive Code

```
public boolean collidesWithBlockSouth (){  
    if(hasBlockOnNextPositionSouth()){  
        return movesSouth() &&  
            (!movesHorizontal() ||  
             !hasBlockPreventingBounceWhenMovingSouth())  
            && isNorthOfBlock();  
    }  
    return false;  
}
```

Best practice:
*Refactor until you
understand the code,
then make your changes*



SPOT Methode lengte; rating 46

Weinig grote uitschieters:
loopt snel terug

spot/SPOT/spot-web/src/main/java/nl/gvb/spot/web/views/KV17View.java

```
225 @Override // NOSONAR: GUI // NOSONAR: GUI
226 protected Table getDetailedTable ( Integer listID ) {
227     Kv17 kv17 = getMessages ( ).get ( listID );
228     Table table = null ;
229     if ( kv17 != null ) {
230         table = new Table ( );
231         table.setStyleName ( "tableDetailMessage" );
232         table.addContainerProperty ( TextConstants.VELD, String.class, null );
233         table.addContainerProperty ( TextConstants.VALUE, String.class, null );
234         addItemToTable ( table, BisonFields.DATAOWNERCODE, kv17.getDataownercode ( ) );
235         addItemToTable ( table, BisonFields.LINEPLANNINGNUMBER, kv17.getLineplanningnumber ( ) );
236         addItemToTable ( table, BisonFields.OPERATINGDAY, String.valueOf ( kv17.getOperatingday ( ) ) );
237         addItemToTable ( table, BisonFields.JOURNEYNUMBER, String.valueOf ( kv17.getJourneynumber ( ) ) );
238         addItemToTable ( table, BisonFields.REINFORCEMENTNUMBER, String.valueOf ( kv17.getReinforcementnumber ( ) ) );
239         if ( ! ( kv17.getMsgtype ( ).equalsIgnoreCase ( KV17MessageTypes.CANCEL.name ( ) )
240             || kv17.getMsgtype ( ).equalsIgnoreCase ( KV17MessageTypes.RECOVER.name ( ) ) ) ) {
241             addItemToTable ( table, BisonFields.USERSTOPCODE, kv17.getUserstopcode ( ) );
242             addItemToTable ( table, BisonFields.PASSEGESEQUENCENUMBER,
243                 String.valueOf ( kv17.getPassagesequencenumber ( ) ) );
244         }
245         if ( kv17.getMsgtype ( ).equalsIgnoreCase ( KV17MessageTypes.CHANGEPASSTIMES.name ( ) ) ) {
246             addItemToTable ( table, BisonFields.DESTINATIONCODE, String.valueOf ( kv17.getDestinationcode ( ) ) );
247             addItemToTable ( table, BisonFields.DESTINATIONNAME50, String.valueOf ( kv17.getDestinationname50 ( ) ) );
248         }
249         if ( kv17.getMsgtype ( ).equalsIgnoreCase ( KV17MessageTypes.CHANGEDESTINATION.name ( ) ) ) {
250             addItemToTable ( table, BisonFields.DESTINATIONNAME16, kv17.getDestinationname16 ( ) );
251             addItemToTable ( table, BisonFields.DESTINATIONDETAILS, kv17.getDestinationdetails ( ) );
252             addItemToTable ( table, BisonFields.DESTINATIONDISPLAY, kv17.getDestinationdisplay ( ) );
253         }
254         if ( kv17.getMsgtype ( ).equalsIgnoreCase ( KV17MessageTypes.CHANGEPASSTIMES.name ( ) ) ) {
255             addItemToTable ( table, BisonFields.TARGETARRIVALTIME, String.valueOf ( kv17.getTargetarrivaltime ( ) ) );
256         }
257     }
258 }
```

Repeterende
structuur valt
op (1)

Top 100 Function Size In LOC

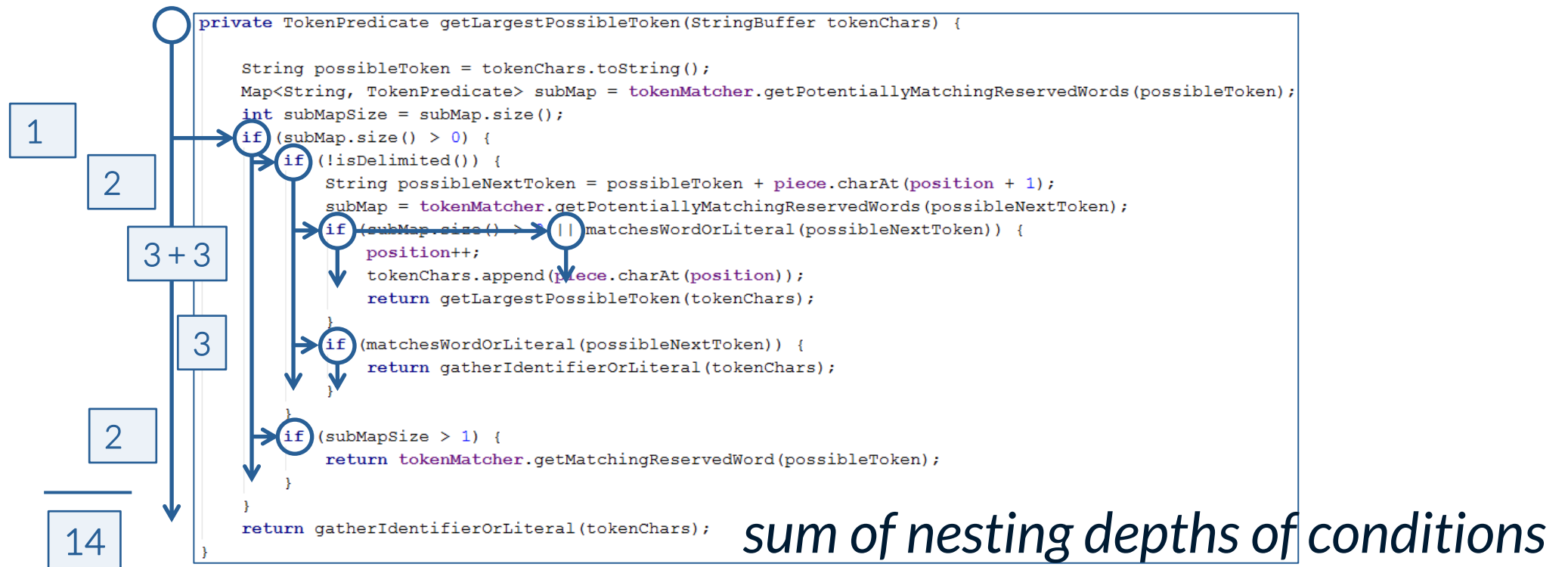
```
66 logToFile( NotifyPackage < ? > s, String fileLocation ) in HandleNotification
45 getDetailedTable( Integer listID ) in KV17View.java (48 lines)
45 getDetailedTable( Integer listID ) in KV6View.java (48 lines)
43 writeToFile( String collectionname, LocalDate checkDate, SortedSet < ? > s ) in HandleNotification
40 createForEachLineOrQuay( SubscribePackage sp, String filename ) in MainCreator.java (41 lines)
39 create( String routes ) in MainCreator.java (41 lines)
37 getDetailedTable( Integer listID ) in KV15View.java (40 lines)
36 createKV15( STOPMESSAGEType sm ) in Kv15Builder.java (38 lines)
35 baseMonitoredCallOnOnwardCall( AbstractMonitoredCallStructure s, String filename ) in MainCreator.java (41 lines)
34 translateToEt( DatedVehicleJourneyStructure dvjPT, DatedTimetable dt ) in MainCreator.java (41 lines)
34 sendStructure( final SiriProducerStructure s, final SiriProducerDocPo s ) in MainCreator.java (41 lines)
33 configure() in BaseRoutes.java (36 lines)
32 getDetailedTable( Integer listID ) in SiriETView.java (35 lines)
32 updateTabpanel() in UsersView.java (34 lines)
31 addFileToTable( Kv1 kv1, String fileName, boolean isKV1, int index ) in MainCreator.java (41 lines)
30 updatePanel( ClickEvent event ) in BisonView.java (33 lines)
30 createWindow( Integer itemId ) in MessagesView.java (32 lines)
30 handle() in VDBRead.java (32 lines)
30 create( Kv15 kv15, Situation situation, SupportedLanguage language ) in MainCreator.java (41 lines)
30 configure() in GenerateReceiveMessageConsumerRoute.java (33 lines)
30 convert( String type ) in ConvertDay.java (32 lines)
```



Method complexity

Rationale

- Too many execution paths impede testability
- Deeply nested code is hard to understand



SPOT: Methode complexiteit

The screenshot displays the SPOT tool interface. On the left, a list of packages is shown, with `NotifyStopMonitoringPackage.java` selected. Below this, a table titled "Top 100 Function Nesting Complexity" lists various methods. Two methods are highlighted with a blue box: `idString() in NotifyStopMonitoringPackage.java (23 lines)` and `idString() in NotifyVehicleMonitoringPackage.java (23 lines)`. An arrow points from the first method to the right-hand pane, which shows the source code of `idString()` in `NotifyStopMonitoringPackage.java`. The code is highlighted in yellow and shows a complex nested structure with multiple loops and conditionals.

Top 100 Function Nesting Complexity

Rank	Method	Lines
27	<code>idString() in NotifyStopMonitoringPackage.java</code>	23 lines
27	<code>idString() in NotifyVehicleMonitoringPackage.java</code>	23 lines
26	<code>process() in DeliverFileState.java</code>	33 lines
25	<code>sendStructure(final SiriProducerStructure s , final SiriProducerDocPort</code>	
24	<code>fillList(final String line , final List < GoviResponse > responses) in GoviRe</code>	
24	<code>translate(LineRefStructure lineref , DirectionRefStructure directionref)</code>	
24	<code>writeToFile(String collectionname , LocalDate checkDate , SortedSet < S</code>	
23	<code>getCollection(final MongoDBCollectionType collection) in MongoDBSer</code>	
20	<code>changeSequenceNumber(KV17CvInfoType kv17) in KV17toXmlAsync</code>	
20	<code>checkForConstistencies(final StopMonitoringDeliveryStructure smd) in</code>	
20	<code>executeQuerySingle(SpotQuery < T > query) in AbstractBasePersist.jav</code>	
19	<code>createBtnClickListener() in UsersView.java</code>	23 lines
19	<code>getDetailedTable(Integer listID) in KV6View.java</code>	48 lines
17	<code>idString() in NotifyGeneralMessagePackage.java</code>	19 lines
17	<code>getDetailedTable(Integer listID) in KV17View.java</code>	48 lines

```
spot/SIRI/siri-interfaces/src/main/java/nl/gvb/siri/interfaces/packages/NotifyStopMonitoringPa
28 final WsStopMonitoringNotificationStructure smn = new WsStop
29 smn.setNotification ( getNotification ( ) ) ;
30 smn.setServiceDeliveryInfo ( getInfo ( ) ) ;
31 smn.setSiriExtension ( getExtension ( ) ) ;
32 return JAXBUtil.marshallAnySiriObjectToString ( smn , WsStop
33 }

35
36 @Override
37 public String idString ( ) {
38     ArrayList < String > dvjs = new ArrayList < > ( ) ;
39     if ( getNotification ( ) != null ) {
40         for ( StopMonitoringDeliveryStructure vmd : getNotifica
41             for ( MonitoredStopVisitStructure va : vmd.getMoni
42                 if ( va.getMonitoredVehicleJourney ( ) != null
43                     if ( va.getMonitoredVehicleJourney ( ).ge
44                         dvjs.add ( Refs.getValue ( va.getMoni
45                     }
46                 }
47             for ( MonitoredStopVisitCancellationStructure vac :
48                 if ( vac.getVehicleJourneyRef ( ) != null ) {
49                     if ( vac.getVehicleJourneyRef ( ).getDate
50                         dvjs.add ( Refs.getValue ( vac.getVel
51                     }
52                 }
53             }
54         }
55     }
56     return "SM:" + dvjs.toString ( ) ;
57 }
58 }
```

Interface stability

Rationale:

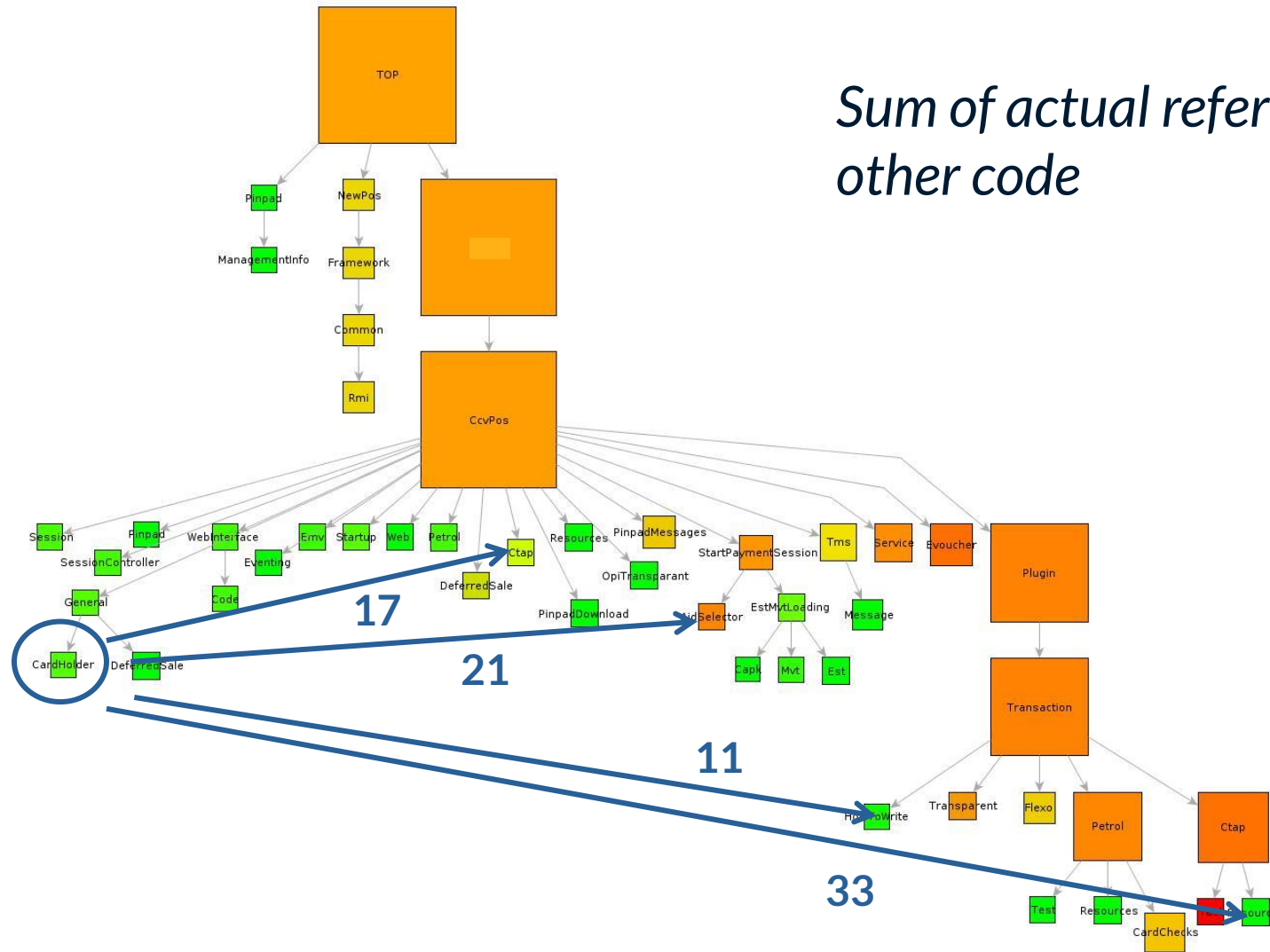
- Functions should have limited responsibility
- Data should be encapsulated

```
public boolean commitSaveInvoiceLine( int mode, boolean newArticle, boolean saveArticleSupplier, boolean newDirectInvoiceLine,
boolean particleChanged, boolean particleFound, boolean changeOrderReceived,
IInvoice invoice, IInvoiceLine invoiceline, IArticle article, IOrder order,
IOrderLine orderLine, IOrderLineReceive receive, IParticle prevPArticle, IParticle particle,
TreeMap<String, IGroupOrder> groupOrders, IArticleSupplier articleSupplier)
```

Invocations
hard to
understand

Difficult to test all
combinations

Dependency volume



SPOT: dependency volume; rating 55

spot/SPOT/spot-web/src/main/java/nl/gvb/spot/web/views/UsersView.java

```
321     }

322     private void updateTechnicalUser ( ) {
323         if ( tblTechnicalUsers != null && ! tblTechnicalUsers . getContainerDataSource ( ) . getItemIds (
324             for ( int i = 0 ; i < tblTechnicalUsers . getContainerDataSource ( ) . getItemIds ( ) . size (
325                 Item item = tblTechnicalUsers . getItem ( i ) ;
326                 USERPERSIST . getUserbyID ( getID ( item ) ) . ifPresent ( user - > {
327                     user . setEnabled ( getBoolean ( item , TextConstants . ACTIVE ) ) ;
328                 try {
329                     USERPERSIST . updateUser ( user ) ;
330                 } catch ( InterruptedException e ) {
331                     Thread . currentThread ( ) . interrupt ( ) ;
332                 }
333                 USERPERSIST . stop ( ) ;
334             } ) ;
335         }
336         SpotUI . showNotification ( ValoTheme . NOTIFICATION_SUCCESS , TextConstants . SUCCES ,
337             TextConstants . MSGUSERSUPDATED ) ;
338     }
339 }

340

341 private void updateProcessUser ( ) {
342     if ( tblProcessUsers != null && ! tblProcessUsers . getContainerDataSource ( ) . getItemIds ( ) . i
343         for ( int i = 0 ; i < tblProcessUsers . getContainerDataSource ( ) . getItemIds ( ) . size ( ) ;
344             Item item = tblProcessUsers . getItem ( i ) ;
345             USERPERSIST . getUserbyID ( getID ( item ) ) . ifPresent ( user - > {
346                 user . setEnabled ( getBoolean ( item , TextConstants . ACTIVE ) ) ;
347             }
```

Code raakt via call
chains ongemerkt veel
code elders



Dependency span



SPOT dependency span

spot/SIRI/siri-plugin-receivemessages/src/main/java/nl/gvb/siri/plugin/receivemessages/ReceiveMessages.java

```
• 1 package nl . gvb . siri . plugin . recevemessages ;

3 import java . io . File ;
4 import java . io . FileNotFoundException ;
5 import java . io . PrintWriter ;

7 import javax . xml . namespace . QName ;

9 import org . apache . activemq . ActiveMQConnectionFactory ;
10 import org . apache . activemq . camel . component . ActiveMQComponent ;
11 import org . apache . camel . component . cxf . CxfComponent ;
12 import org . apache . camel . component . cxf . CxfEndpoint ;
13 import org . apache . camel . impl . DefaultCamelContext ;
14 import org . apache . camel . impl . SimpleRegistry ;
15 import org . apache . cxf . interceptor . LoggingInInterceptor ;
16 import org . apache . cxf . transport . common . gzip . GZIPInInterceptor ;
17 import org . apache . cxf . transport . common . gzip . GZIPOutInterceptor ;
18 import org . osgi . framework . BundleException ;

20 import nl . gvb . siri . interfaces . Plugin ;
21 import nl . gvb . siri . interfaces . SiriBase ;
22 import nl . gvb . siri . interfaces . packages . SiriPackage ;
23 import nl . gvb . siri . interfaces . types . ServiceName ;
24 import nl . gvb . siri . interfaces . types . SiriConstants ;
25 import nl . gvb . siri . utils . SiriConfig ;
26 import nl . gvb . spot . utils . Config ;
27 import nl . gvb . spot . utils . ConfigMap ;
28 import nl . gvb . spot . utils . ManagedBundle ;
29 import nl . gvb . spot . utils . SpotLog ;
30 import uk . org . siri . wsdl . cons . SiriConsumerDocPort ;
31 import uk . org . siri . wsdl . prod . SiriProducerDocPort ;
```

Grotere classes hebben veel
afhankelijkheden

Maar die zijn over het algemeen
niet heel divers



Best practice metrics



Best practice metrics

- Error handling
 - Upper level catches
 - Empty catches
 - Illegal catches
- Modularity
 - Cyclic references
- Unfinished work
 - TODO
 - FIXME
- Customer / context specific
 - SQL concatenation
 - Copyright/licensing issues
 - ...



SPOT Illegal catches

- 5 Illegal catches (upper level, empty catch)
- Alle in HandleNotification

Illegal Catch
ementations/HandleNotification.java)
ementations/HandleNotification.java)
ementations/HandleNotification.java)
ementations/HandleNotification.java)
plementations/HandleNotification.java)

spot/SIRI/siri-utils/src/main/java/nl/gvb/siri/utils/default_implementations/HandleNotification.java

```
56         } catch ( Exception e ) { }
57         filename = time + "_-ET-" + id ;
58         write ( fileLocation , filename , s . toSoapString ( "NotifyEstimatedTimetable" ) ) ;
59     } else if ( s instanceof NotifyGeneralMessagePackage ) {
60         try {
61             id = ( ( NotifyGeneralMessagePackage ) s )
62                 . getNotification ( )
63                 . getGeneralMessageDelivery ( ) . get ( 0 )
64                 . getGeneralMessage ( ) . get ( 0 )
65                 . getInfoMessageIdentifier ( ) . getValue ( ) ;
66         } catch ( Exception e ) { }
67         filename = time + "_-GM-" + id ;
68         write ( fileLocation , filename , s . toSoapString ( "NotifyGeneralMessage" ) ) ;
69     } else if ( s instanceof NotifyProductionTimetablePackage ) {
```

Actie kan om aantal redenen een exceptie opleveren, inclusief een programmeerfout



SPOT Upper level catches

- 60 upper level catches
- Leeuwendeel daarvan verklaarbaar; in stop / start (service/plugin) methodes

```
spot/SPOT/spot-web/src/main/java/nl/gvb/spot/web/views/DegradedModeView.java
98     private List < Object [ ] > getLinesAndFill ( ) {
99         try {
100             FindIterable < Document > lines = persister . quer
101             List < Object [ ] > rows = new ArrayList < > ( ) ;

103             for ( Document line : lines ) {
104                 rows . add ( new Object [ ] { addButton ( lin
105                     "lijn " + line . getDouble ( LINE_NUMI
106             }
107             return rows ;
108         } catch ( Exception e ) {
109             LOG . error ( "Couldn't find any lines" ) ;
110         }

112         return Collections . emptyList ( ) ;
113     }
```

Logging aanwezig

Logging is overigens
consistent (Slf4J)



SPOT logging

spot/SPOT/spot-web/src/main/java/nl/gvb/spot/web/views/DegradedModeView.java

```
20 import nl.gvb.spot.mdb.MongoDbService ;
21 import nl.gvb.spot.mdb.entity.MongoDbCollectionType ;
22 import nl.gvb.spot.utils.NoConnectionException ;
23 import nl.gvb.spot.utils.SpotLog ;
24 import nl.gvb.spot.web.SpotUI ;
25 import nl.gvb.spot.web.TextConstants ;

27 @SuppressWarnings ( "serial" )
28 public class DegradedModeView extends SpotView {
29     private static final String LINE_NUMBER = "lineNumber" ;
30     private static final String FIRSTCOLUMN = "FIRSTCOLUMN" ;
31     private static final String SECONDCOLUMN = "SECONDCOLUMN" ;

33     private static final SpotLog LOG = new SpotLog ( "DegradedModeView" ) ;

35     private static MongoDbService persister ;

37     public DegradedModeView ( ) {
38         if ( persister == null ) {
39             persister = new MongoDbService ( ) ;
40             persister.start ( ) ;
41         }
42     }

44     private void setDegradedModeStatus ( String lineNumber ) throws NoConnectionException {
45         LOG.info ( "----- DEGRADED MODE STATUS CHANGED FOR LINE: " + lineNumber + " -----" ) ;
    }
```

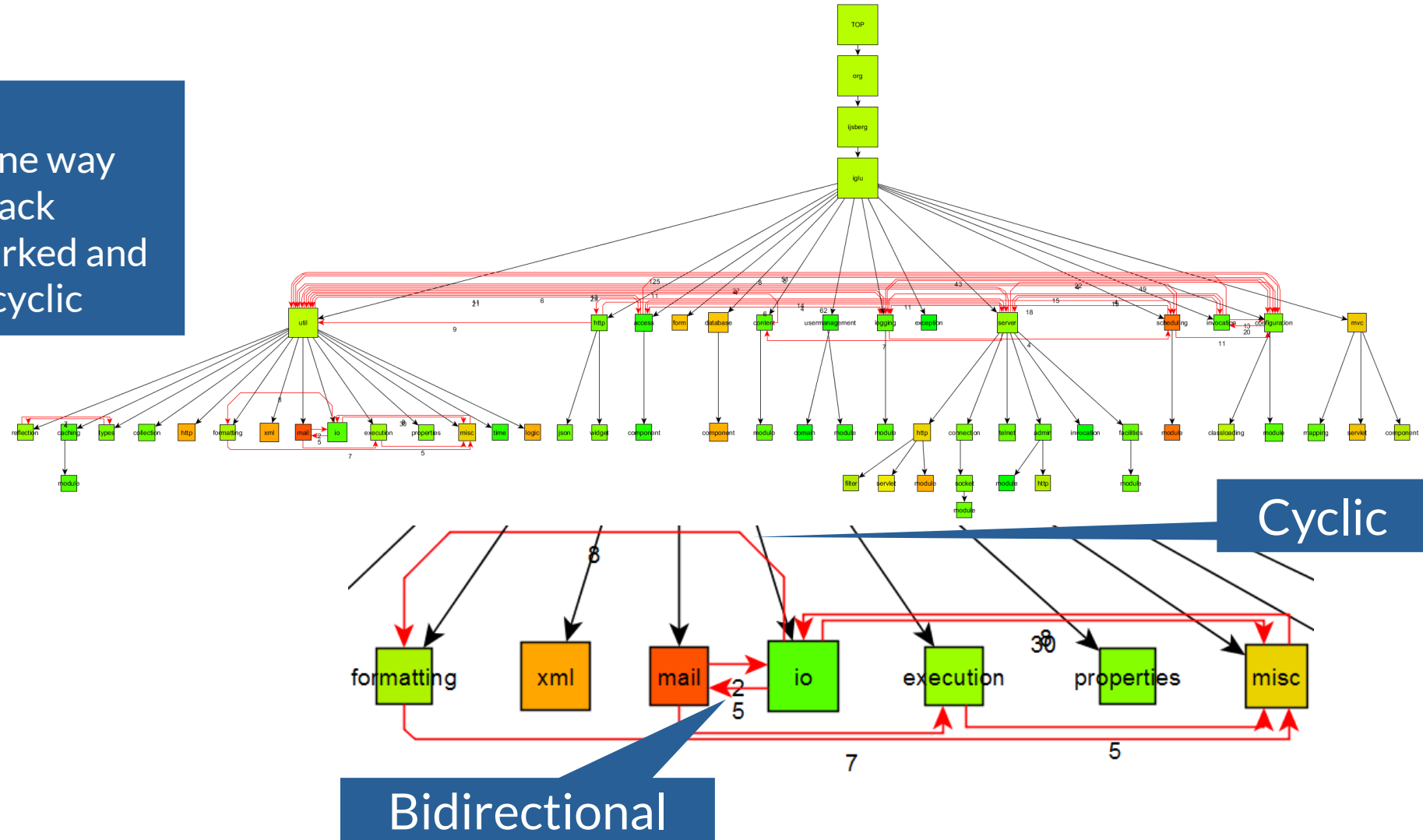
SpotLog

Wrapper voor Slf4J

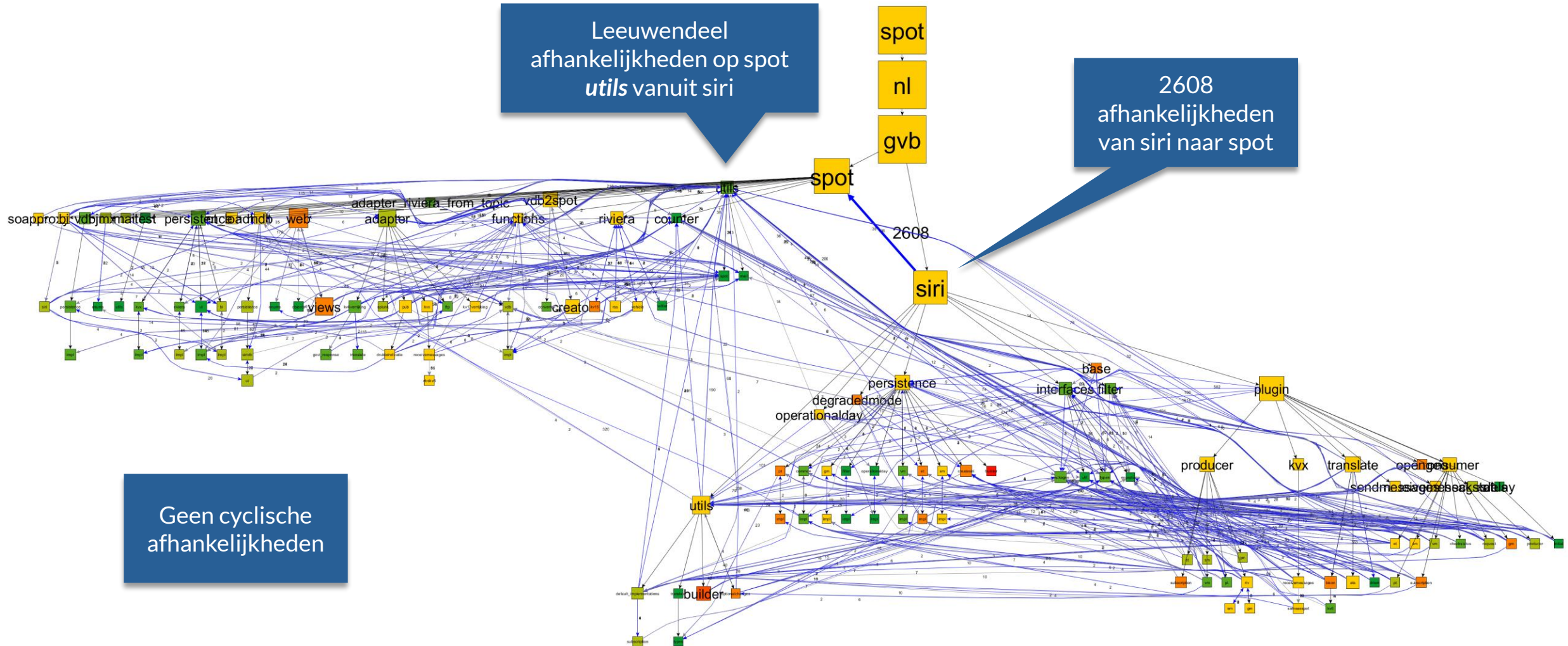


Cyclic dependencies

Metric:
2 pointing one way
5 pointing back
Means 2 marked and
counted as cyclic



SPOT dependency graph



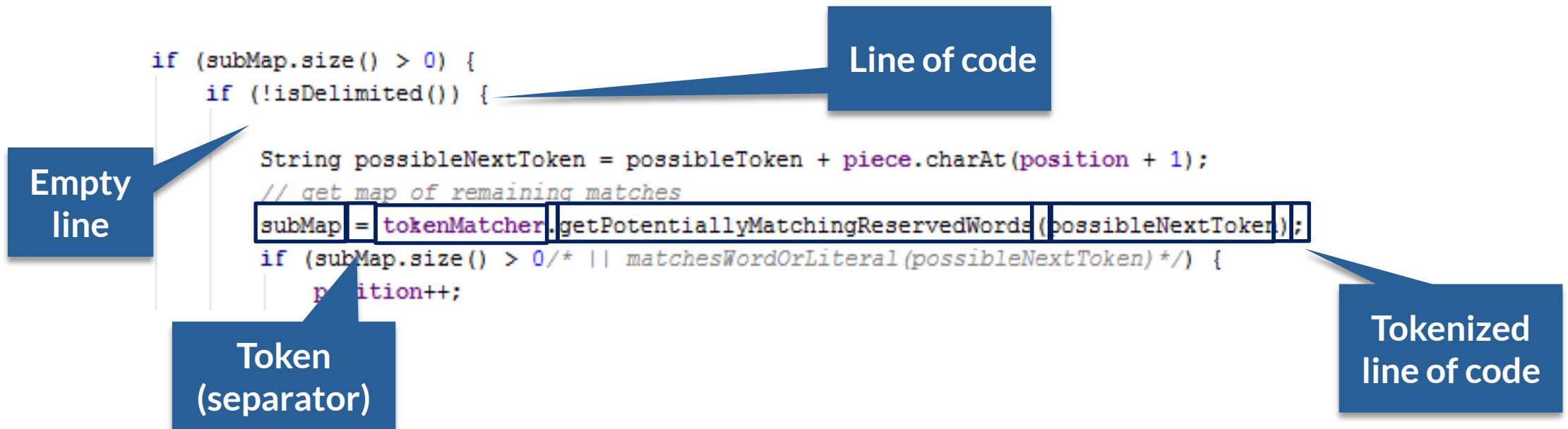
Volume metrics



Volume is measured in total tokens

Rationale:

- Analysis of a bigger system consumes more labor time
- More code results in higher maintenance costs



Duplication

Rationale:

- Duplication is a signal of lack of abstraction
- Duplication multiplies maintenance effort

```
public String getText(String categoryKey, String key) {  
    Properties texts = textsByCategory.get(categoryKey);  
    String text = null;  
    if(texts != null) {  
        text = texts.getProperty(key);  
    }  
    if(text == null) {  
        text = getText(key);  
    }  
    return text;  
}  
  
public String getText(String categoryKey, String key,  
    String defaultText) {  
    Properties texts = textsByCategory.get(categoryKey);  
    String text = null;  
    if(texts != null) {  
        text = texts.getProperty(key);  
    }  
    if(text == null) {  
        text = defaultText;  
    }  
    return text;  
}
```

84 tokens marked as duplicate
whereof 42 tokens are redundant



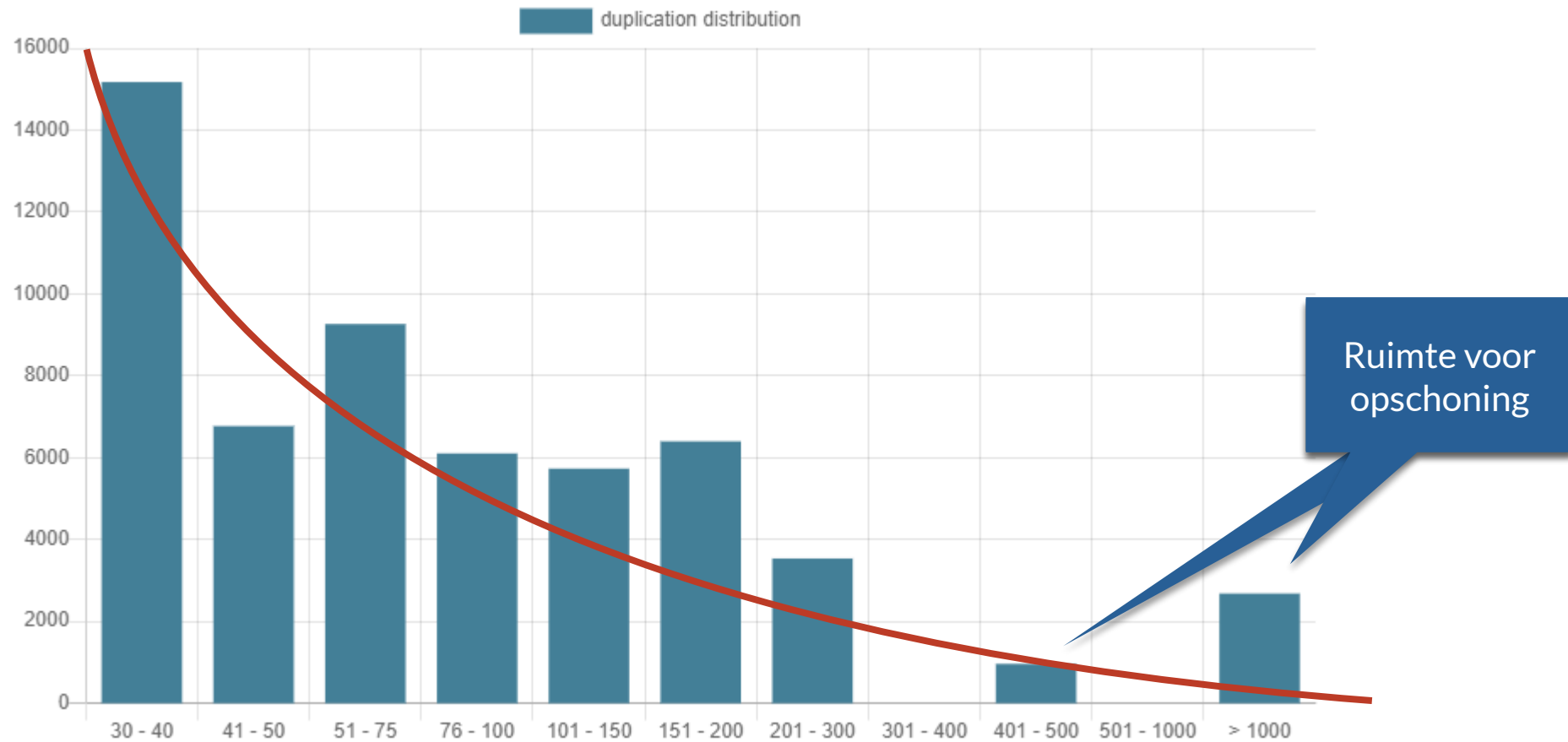
SPOT volume & duplicatie

- Volume 50 kLOC, kwalificatie klein tot middelgrote applicatie
- 17 % van code aangemerkt als duplicaat
- 45 % van de code aangemerkt als onderdeel patroonduplicatie
- Aantal doorlooppaden door methods 7.640
- 17.839 LOC test code (1 regel test code op op 2,8 regels productie code)
- 1.558 afgeteste situaties (1 assert op 5 doorlooppaden)
- 280 bestanden

VOLUME	
Volume In Lines Of Code	49470
Volume In Lines	62384
Lines Of Comment	5003
Volume In Tokens	398274
Duplicate Tokens Percentage	17.2 %
Duplicate Pattern Tokens Percentage	45.5 %



Duplicatie distributie



Top 10 duplicaten

Literal Duplicates Top 10

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

1340 pieces in 2 fragments

lines 50-222 in spot/SIRI/siri-plugin-kvx-receivemessages/src/main/java/nl/gvb/siri/plugin/kvx/receivemessages/sameasspot/EBStoKV17.java

lines 50-222 in spot/SPOT/spot-adapter-kvx-receivemessages/src/main/java/nl/gvb/spot/adapter/kvx/receivemessages/EBStoKV17.java

728 pieces in 2 fragments

lines 327-403 in spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/batch/VmIntoSql.java

lines 294-370 in spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/single/loginto/LogVmIntoVdbMsSqlServer.java

658 pieces in 2 fragments

lines 237-314 in spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/batch/EtIntoSql.java

lines 202-279 in spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/single/loginto/LogEtIntoVdbMsSqlServer.java

589 pieces in 2 fragments

lines 120-181 in spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/batch/PtIntoSql.java

lines 94-155 in spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/single/loginto/LogPtIntoVdbMsSqlServer.java

481 pieces in 2 fragments

lines 28-92 in spot/SIRI/siri-plugin-kvx-receivemessages/src/main/java/nl/gvb/siri/plugin/kvx/receivemessages/sameasspot/VLSMtoKV17.java

lines 27-91 in spot/SPOT/spot-adapter-kvx-receivemessages/src/main/java/nl/gvb/spot/adapter/kvx/receivemessages/VLSMtoKV17.java

481 pieces in 2 fragments

lines 203-252 in spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/batch/VmIntoSql.java

lines 172-221 in spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/single/loginto/LogVmIntoVdbMsSqlServer.java

“sameasspot”
Rationale achter
duplicatie?



Voorbeeld duplicatie

Repeterende
structuur valt
op (2)

spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/batch/EtIntoSql.java

```
45     String inOrOut )  
46     {  
47         Map map = new HashMap < > ( ) ;  
48         map . put ( "Id" , Integer . toString ( Id ) ) ;  
  
50         safeputTime ( map , "ResponseTimestamp_ServiceDeliveryInfo",  
51                     x - > ( ( NotifyEstimatedTimetablePackage ) x ) . getIn  
52                     x - > ( ( ProducerResponseEndpointStructure ) x ) . get  
53                     ) ;  
54         safeputString ( map , "ProducerRef" , notification ,  
55                       x - > ( ( NotifyEstimatedTimetablePackage ) x ) . getIn  
56                       x - > ( ( ProducerResponseEndpointStructure ) x ) . get  
57                       x - > ( ( ParticipantRefStructure ) x ) . getValue ( )  
58                       ) ;  
59         safeputString ( map , "ResponseMessageIdentifier" , notific  
60                       x - > ( ( NotifyEstimatedTimetablePackage ) x ) . getIn  
61                       x - > ( ( ProducerResponseEndpointStructure ) x ) . get  
62                       x - > ( ( MessageQualifierStructure ) x ) . getValue ( )  
63                       ) ;  
64         safeputString ( map , "EstimatedTimetableDelivery_version" , delivery ,  
65                       x - > ( ( EstimatedTimetableDeliveryStructure ) x ) . getVersion ( )  
66                       ) ;  
67         safeputTime ( map , "ResponseTimestamp_EstimatedTimetableDelivery" , delivery ,  
68                       x - > ( ( EstimatedTimetableDeliveryStructure ) x ) . getResponseTimestamp ( )  
69     }  
70 }
```

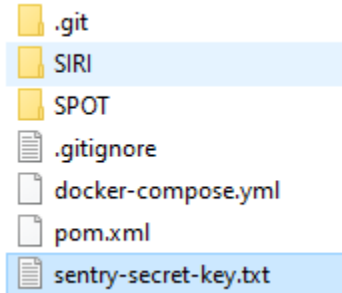
references

EtIntoSql.java (DUPLICATE_START)
45 spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/a
PtIntoSql.java (DUPLICATE_START)
40 spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/a
VmIntoSql.java (DUPLICATE_START)
54 spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/a

Overige bevindingen



Secrets



Hoe worden “secrets”
bewaard?

spot/SPOT/spot-utils/src/main/java/nl/gvb/spot/spot/SpotConfig.java

```
15 public static final Config < String > ACTIVEMQ_PASSWORD = new Config < > ( "ActiveMQPassword", String.class, true );
16 public static final Config < String > ACTIVEMQ_URL = new Config < > ( "ActivemqUrl", String.class );
17 public static final Config < String > ACTIVEMQ_USERNAME = new Config < > ( "ActiveMQUsername", String.class, true );
18 public static final Config < String > BI_SIRI_SCHEDULE = new Config < > ( "SiriSchedule", String.class );
19 public static final Config < String > BI_CLEAN_LOG_SCHEDULE = new Config < > ( "CleanLogSchedule", String.class );
20 public static final Config < String > BI_DAILY_SCHEDULE = new Config < > ( "DailySchedule", String.class );
21 public static final Config < String > BI_FILEPATH = new Config < > ( "BiFilepath", String.class );
```



Gegenereerde code? uk.org.siri?

```
spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/batch/VmIntoSql.java
43     import uk.org.siri.siri.VehicleMonitoringDeliveryStructure ;
44     import uk.org.siri.siri.VehicleMonitoringRefStructure ;
45     import uk.org.siri.siri.VehicleRefStructure ;

47     import static nl.gvb.siri.tools.analyse.single.loginto.SafePut.* ;

49     public class VmIntoSql {

51         public Map < String , String > VehicleMonitoring_IN_MessagesRow (
52             int Id ,
53             NotifyVehicleMonitoringPackage notification ,
54             VehicleMonitoringDeliveryStructure delivery , String inOrOut )
55         {
56             Map map = new HashMap < > ( ) ;
57             map.put ( "Id" , Integer.toString ( Id ) ) ;
58             safeputTime ( map , "ResponseTimestamp_ServiceDeliveryInfo" , notification ,
59                 x -> ( ( NotifyVehicleMonitoringPackage ) x ).getInfo ( ) ,
60                 x -> ( ( ProducerResponseEndpointStructure ) x ).getResponseTimestamp ( )
61             ) ;
62             safeputString ( map , "ProducerRef" , notification ,
63                 x -> ( ( NotifyVehicleMonitoringPackage ) x ).getInfo ( ) ,
64                 x -> ( ( ProducerResponseEndpointStructure ) x ).getProducerRef ( ) ,
65                 x -> ( ( ParticipantRefStructure ) x ).getValue ( )
66             ) ;
67             safeputString ( map , "ResponseMessageIdentifier" , notification ,
68                 x -> ( ( NotifyVehicleMonitoringPackage ) x ).getInfo ( ) ,
69                 x -> ( ( ProducerResponseEndpointStructure ) x ).getResponseMessageIdentifier ( ) ,
70                 x -> ( ( MessageQualifierStructure ) x ).getValue ( )
71             ) ;
```



Gegenereerde code in "entity"

spot/SPOT/base-persister/src/main/java/nl/gvb/spot/entity/Riviera.java

```
1 package nl.gvb.spot.entity ;

3 import static javax.persistence.GenerationType.IDENTITY ;

4 // Generated 11-sep-2015 9:21:49 by Hibernate Tools 4.0.0
6 import java.time.OffsetDateTime ;

8 import javax.persistence.Column ;
9 import javax.persistence.Entity ;
10 import javax.persistence.FetchType ;
11 import javax.persistence.GeneratedValue ;
12 import javax.persistence.Id ;
13 import javax.persistence.JoinColumn ;
14 import javax.persistence.ManyToOne ;
15 import javax.persistence.Table ;

/**
 * Riviera generated by hbm2java
 */

21 @Entity
22 @Table ( name = "riviera" )
23 public class Riviera {
```

Navigation

- src
- main
- java
- nl
- gvb
- siri
- entity
 - SiriAccount.java
 - SiriAffectsscope.java
 - SiriDatedvehiclejourney.java
 - SiriGeneralmessage.java
 - SiriMonitoredcall.java
- + persistence
- + siri-plugin-checkstatus-consumer
- + siri-plugin-checkstatus-producer
- + siri-plugin-delay-initial
- + siri-plugin-kvx-receivemessages
- + siri-plugin-opengeo
- + siri-plugin-receivemessages

spot/SIRI/siri-persister/src/main/java/nl/gvb/siri/entity/SiriC

```
1 package nl.gvb.siri.entity ;

3 import static javax.persistence.GenerationType.IDENTITY ;

4 // Generated 24-aug-2015 14:16:07 by
6 import java.time.OffsetDateTime ;
7 import java.util.Optional ;

9 import javax.persistence.Column ;
10 import javax.persistence.Entity ;
11 import javax.persistence.EnumType ;
12 import javax.persistence.Enumerated ;
13 import javax.persistence.FetchType ;
14 import javax.persistence.GeneratedValue ;
15 import javax.persistence.Id ;
16 import javax.persistence.JoinColumn ;
```



NOSONAR

spot/SPOT/base-persister/src/main/java/nl/gvb/spot/entity/RssItem.java

```
63  
64  
65     public void setTitle ( String title ) {  
66         this.title = title ;  
67     }  
  
69     @ Column ( name = "Description", nullable = false , length = 2048 ) // NOSONAR // NOSONAR // NOSONAR  
70     public String getDescription ( ) {  
71         return description ;  
72     }
```



Apache Camel

spot/SPOT/spot-counter/src/main/java/nl/gvb/spot/counter/CounterProducer.java

```
● 1    package nl . gvb . spot . counter ;

      3    import org . apache . camel . Exchange ;
      4    import org . apache . camel . impl . DefaultProducer ;

/**
 * The counter producer.
 *
 * @author geerts
 * @see {@link} http://michalwarecki.blogspot.nl/2012/08/monitoring-and-managing-apache-camel.html
 */
      12    class CounterProducer extends DefaultProducer {
      13        private final Counter counter ;

● 15        CounterProducer ( CounterEndpoint endpoint ) {
      16            super ( endpoint ) ;
      17            counter = new Counter ( endpoint . getTime ( ) , endpoint . getName ( ) ) ;
      18        }
```



Illustratie Upper level & empty catches

spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/deserialize/CSVWriter.java

```
45     }

47     public static boolean generateCSV ( File csvFileName , Object [ ] data ) {
48         FileWriter fw = null ;
49         try {
50             fw = new FileWriter ( csvFileName ) ;
51             if ( ! csvFileName . exists ( ) )
52                 csvFileName . createNewFile ( ) ;
53             fw . write ( produceCsvData ( data ) ) ;
54             fw . flush ( ) ;
55         } catch ( Exception e ) {
56             System . out . println ( "Error while generating csv from data. Error message : " + e . getMessage ( ) ) ;
57             e . printStackTrace ( ) ;
58             return false ;
59         } finally {
60             if ( fw != null ) {
61                 try {
62                     fw . close ( ) ;
63                 } catch ( Exception e ) {
64                 }
65                 fw = null ;
66             }
67         }
68         return true ;
69     }
70 }
```



Test code in production code?

spot/SIRI/siri-tools/src/main/java/nl/gvb/siri/tools/analyse/TestData.java

```
1 package nl.gvb.siri.tools.analyse ;

3 public class TestData {

4     // WsVehicleMonitoringNotificationStructure vehicleMonitoringDeliveryStructure =
5     // JAXBUtil.unmarshalStringToSiriObject(getData(), WsVehicleMonitoringNotificationStructure.class);
6     // JAXBUtil.unmarshalStringToSiriObject();
7     public String getETData ( ) {
8         return "<soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" xmlns:wsdl=\""
9             + "<ServiceDeliveryInfo>\n"
10             + "        <ns2:ResponseTimestamp>2020-06-10T00:01:37.1621194+02:00</ns2:ResponseTimestamp>\n"
11             + "        <ns2:ProducerRef>GVB:DRIS</ns2:ProducerRef>\n"
12             + "        <ns2:ResponseMessageIdentifier>fb15812d-72c8-43b8-9544-975147dda603</ns2:ResponseMessage
13             + "</ServiceDeliveryInfo>\n"
14             + "<Notification>\n"
15             + "        <ns2:EstimatedTimetableDelivery version=\"2.0:NL-GVB-1.0\">\n"
16             + "            <ns2:ResponseTimestamp>2020-06-10T00:01:37.139+02:00</ns2:ResponseTimestamp>\n"
17             + "            <ns2:SubscriberRef>GVB:SPOTACC</ns2:SubscriberRef>\n"
18             + "            <ns2:SubscriptionRef>GVB:Subscription:ET:000341</ns2:SubscriptionRef>\n"
19             + "            <ns2:Status>true</ns2:Status>\n"
20             + "            <ns2:EstimatedJourneyVersionFrame>\n"
21             + "                <ns2:RecordedAtTime>2020-06-10T00:01:36.873+02:00</ns2:RecordedAtTime>\n"
22             + "                <ns2:EstimatedVehicleJourney>\n"
```



Hard coded URLs?

```
spot/SIRI/siri-operational-day/src/main/java/nl/gvb/siri/operationalday/ODRoute.java
51     }

53     private void startRoutes ( ) {
54         from ( "quartz2:operational_day?cron=" + generateSchedule ) .routeId ( "Operational day - Quartz" )
55         .setHeader ( DAYS_TO_DO , constant ( daysInAdvance ) )
56         .to ( DIRECT_GENERATE_OD ) ;

        // @formatter:off
59         from ( "jetty:http://0.0.0.0:1236" ) .routeId ( "Operational day - Jetty" )
60         .choice ( ) .when ( header ( "days_todo" ) .isNull ( ) )
61         .setHeader ( DAYS_TO_DO , constant ( daysInAdvance ) )
62         .otherwise ( )
63         .setHeader ( DAYS_TO_DO , header ( "days_todo" ) )
64         .end ( )
65         .wireTap ( DIRECT_GENERATE_OD )
66         .transform ( constant ( "Operational day has been started" ) ) ;

67         // @formatter:on
68     }

70     private void cleanRoutes ( ) {
71         from ( "jetty:http://0.0.0.0:1236/clean" ) .routeId ( "Operational day - Clean - Jetty" )
72         .wireTap ( DIRECT_CLEAN_OD )
73         .transform ( constant ( "Operational day cleaning has been started" ) ) ;
74     }
```



Oldskool JDBC?

spot/SPOT/base-persister/src/main/java/nl/gvb/spot/persistence/ExecuteSQL.java

```
38         if ( connection == null ) {
39             LOG.info ( "KARAF LOGING URL FAILED" );
40             connection = new MariaDbDataSource ( cd.getUrl ( ) ).getConnection ( cd.getUser ( ), cd.getPassword ( ) );
41         }
42         return connection ;
43     }

44     • public void close ( ) throws SQLException {
45         if ( connection != null && ! connection.isClosed ( ) ) {
46             connection.close ( ) ;
47             connection = null ;
48         }
49     }
50

51     • public void runQuery ( String query ) throws SQLException {
52         if ( connection != null && ! connection.isClosed ( ) ) {
53             try ( Statement st = connection.createStatement ( ) ) {
54                 st.executeUpdate ( query ) ;
55             }
56         } else {
57             LOG.error ( "No connection, please connect first!" ) ;
58         }
59     }
60 }
```



Datum in sleutel?



Titel : Database Design Description (DBDD) SPOT
Kenmerk : SPOT-DBDD Versie : 2.1
Status : Voorlopig

4 Database detailontwerp

▲ 4.1 Algemeen

Elke tabel heeft een id/als 'technische' sleutel. Daarnaast is per tabel aangegeven welke velden samen de 'functionele' sleutel vormen.

Bij de 'relaties' van de tabel is aangegeven of het een 'echte' foreign key (FK) betreft of niet. In het eerste geval verwijst het veld daadwerkelijk naar de id van het record in de andere tabel. Dit is echter niet mogelijk bij verwijzingen naar tabellen waarbij **de geldigheid (veld *FromDate*) onderdeel uitmaakt van de sleutel** terwijl de verwijzende tabel een andere geldigheidsperiode heeft. In dergelijke gevallen bevat de verwijzing alle overige (functionele) sleutelvelden. Het juiste record kan pas worden bepaald a.d.h.v. de relevante datum of periode. Zie ook § 4.2.41.

DBDD 2.5

Opdrachtgever : GVB
Auteur : SPOT-team (InTraffic)
Kenmerk : SPOT-DBDD
Versie : 2.5
Datum : 24-04-2019
Bestandsnaam : SPOT-DBDD_v25
Status : Voorlopig

Tabelstructuur

Kolom	Type	NULL?	PK	Omschrijving
id	BIGINT(20)	N	1	De technische sleutel
PointNumber	VARCHAR(4)	N		Verkorte identificatie
FromDate	DATE	N		Begin van de geldigheid
ToDate	DATE			Einde van de geldigheid (NULL = oneindig)
VersionRef	BIGINT(20)			Versiegegevens (nog niet ingevuld)
Name	VARCHAR(45)			Volledige naam
PrivateCode_VDV	INTEGER			Identificatie in EBS
PrivateCode_ETG	INTEGER			Identificatie in ETG of OVC
ShortStopCode	VARCHAR(8)			Puntafkorting
ShortStopCode_VLSM	VARCHAR(3)			Puntafkorting binnen VLSM
StopType	ENUM	N		Soort halte: 'onstreetBus', 'onstreetTram', 'ferryStop', 'metroStation', 'tramStation', 'coachStation', 'other', 'unknown'
ForBoarding	TINYINT(1)	N		Geeft aan of het een instaphalte kan zijn (vooralnog altijd TRUE)
ForAlighting	TINYINT(1)	N		Geeft aan of het een uitstaphalte kan zijn (vooralnog altijd TRUE)
RequestStop	TINYINT(1)	N		Geeft aan of het een verzoekhalte is (vooralnog altijd FALSE)
TariffZone	INTEGER			Zone
StopPlace	VARCHAR(8)	N		Haltegebied (fysiek)
QuayRef	BIGINT(20)	N		(Default) perrontoewijzing

spot/SPOT/resources/Create Database/Create Database.sql

-- Data exporting was unselected.

-- Dumping structure for table spot_test.stoppoint

```
897 DROP TABLE IF EXISTS `stoppoint`;
898 CREATE TABLE IF NOT EXISTS `stoppoint` (
899   `id` bigint ( 20 ) NOT NULL AUTO_INCREMENT ,
900   `PointNumber` varchar ( 8 ) NOT NULL COMMENT 'punt_code' ,
901   `FromDate` date NOT NULL ,
902   `ToDate` date DEFAULT NULL ,
903   `VersionRef` bigint ( 20 ) DEFAULT NULL ,
904   `Name` varchar ( 45 ) DEFAULT NULL ,
905   `PrivateCode_VDV` int ( 11 ) DEFAULT NULL ,
906   `PrivateCode_ETG` int ( 11 ) DEFAULT NULL ,
907   `ShortStopCode` varchar ( 8 ) DEFAULT NULL ,
908   `ShortStopCode_VLSM` varchar ( 3 ) DEFAULT NULL ,
909   `StopType` enum ( 'onstreetBus' , 'onstreetTram' , 'ferryStop' ,
910   `ForBoarding` tinyint ( 1 ) NOT NULL ,
911   `ForAlighting` tinyint ( 1 ) NOT NULL ,
912   `RequestStop` tinyint ( 1 ) NOT NULL ,
913   `TariffZone` int ( 11 ) DEFAULT NULL ,
914   `StopPlace` varchar ( 8 ) NOT NULL ,
915   `QuayRef` bigint ( 20 ) DEFAULT NULL ,
916   PRIMARY KEY ( `id` ) ,
917   UNIQUE KEY `pk_stoppoint_UNIQUE` ( `PointNumber`,`FromDate` ) ,
918   UNIQUE KEY `id_UNIQUE` ( `id` ) ,
919   KEY `fk_stoppoint_Quay_idx` ( `QuayRef` ) ,
920   CONSTRAINT `fk_stoppoint_Quay` FOREIGN KEY ( `QuayRef` ) REFERENCES
921 ) ENGINE = InnoDB DEFAULT CHARSET = utf8 ;
```

Datedvehiclejourney

Tabelstructuur

Kolom	Type	NULL?	PK	Omschrijving
id	BIGINT(20)	N	1	De technische sleutel
VehicleJourneyRef	BIGINT(20)	N		Bijbehorende rit in de dienstregeling
OperatingDate	DATE	N		Exploitatiedatum
ExternalDatedJourney	INTEGER			Ritnummer in het externe systeem
PreviousCallRef	BIGINT(20)			De halte vóór degene waar de laatst ontvangen punctualiteit op sloeg
OnwardCallRef	BIGINT(20)			De halte ná degene waar de laatst ontvangen punctualiteit op sloeg
Linenummer	INTEGER	N		Bijbehorend lijnnummer
Publiccode	INTEGER	N		Bijbehorende lijnnaam
VehicleJourneyShortName	INTEGER	N		Bijbehorend ritnummer
VehicleStatus	ENUM			Status van het voertuig: 'expected', 'notExpected', 'cancelled', 'assigned', 'signedOn', 'atOrigin', 'inProgress', 'aborted', 'offRoute', 'completed', 'assumedCompleted', 'notRun'
FirstOrLastJourney	ENUM			Is het de eerste of laatste rit van de dag: 'firstServiceOfDay', 'otherService', 'lastServiceOfDay', 'unspecified'
Monitored	TINYINT	N		
Delay	INTEGER	N		Vertraging van het voertuig
ProgressRate	ENUM			Voortgangssnelheid van het voertuig: 'noProgress', 'slowProgress', 'normalProgress', 'fastProgress', 'unknown'
VehicleLocation_RDx	INTEGER	N		X positie van het voertuig (RD coördinaten)
VehicleLocation_RDy	INTEGER	N		Y positie van het voertuig (RD coördinaten)
Direction	INTEGER			Richting
JourneyPattern	VARCHAR15			Rit patroon
TransportMode	ENUM			Modaliteit: 'TRAM', 'BUS', 'METRO', 'WATER', 'OTHER', 'UNKNOWN'
LineName	VARCHAR50			Lijnnaam



spot/SPOT/resources/Create Database/Create Database.sql

-- data exporting was unsuccessful.

-- Dumping structure for table spot_test.datedvehiclejourney

```

117 DROP TABLE IF EXISTS `datedvehiclejourney`;
118 CREATE TABLE IF NOT EXISTS `datedvehiclejourney` (
119   `id` bigint ( 20 ) NOT NULL AUTO_INCREMENT ,
120   `VehicleJourneyRef` bigint ( 20 ) NOT NULL ,
121   `OperatingDate` date NOT NULL ,
122   `ExternalDatedJourney` int ( 11 ) DEFAULT NULL ,
123   `PreviousCallRef` bigint ( 20 ) DEFAULT NULL ,
124   `OnwardCallRef` bigint ( 20 ) DEFAULT NULL ,
125   `TripStatus` enum ( 'unknown' , 'driving' ) NOT NULL DEFAULT 'unknown' ,
126   PRIMARY KEY ( `id` ) ,
127   UNIQUE KEY `id_UNIQUE` ( `id` ) ,
128   KEY `fk_datedvehiclejourney_VehicleJourney_idx` ( `VehicleJourneyRef` ) ,
129   KEY `fk_datedvehiclejourney_PreviousCall_idx` ( `PreviousCallRef` ) ,
130   KEY `fk_datedvehiclejourney_OnwardCall_idx` ( `OnwardCallRef` ) ,
131   CONSTRAINT `fk_datedvehiclejourney_OnwardCall` FOREIGN KEY ( `OnwardCallRef` )
132   REFERENCES `datedvehiclejourney` ( `id` ) ,
133   CONSTRAINT `fk_datedvehiclejourney_PreviousCall` FOREIGN KEY ( `PreviousCallRef` )
134   REFERENCES `datedvehiclejourney` ( `id` ) ,
135   CONSTRAINT `fk_datedvehiclejourney_VehicleJourney` FOREIGN KEY ( `VehicleJourneyRef` )
136   REFERENCES `datedvehiclejourney` ( `id` ) ) ENGINE = InnoDB DEFAULT CHARSET = utf8 ;

```



Documentatie nog niet final

Opdrachtgever GVB
InTraffic kenmerk SPOT SDD versie 2.3

Kenmerk SPOT-SDD-2.3
Versie V2.3
Datum 24-07-2019
Bestandnaam SDD SPOT V2.3_concept
Status Concept

Opdrachtgever : GVB
Auteur : SPOT-team (InTraffic)
Kenmerk : SPOT-DBDD
Versie : 2.5
Datum : 24-04-2019
Bestandnaam : SPOT-DBDD_v25
Status : Voorlopig



Rationale gedocumenteerd

- Ontwerpbeslissingen SPOT
- 1. SPOT wordt ontwikkeld in Java.
- 2. Het ontwikkelplatform is Linux (CentOS 7).
- 3. SPOT maakt gebruik van Java – OSGi.
- 4. De databases zijn MySQL / MariaDB en MongoDB
- 5. SPOT wordt ontwikkeld met behulp van Open Source componenten.
- 6. Voor messaging (JMS) wordt ActiveMQ gebruikt.
- 7. De applicatieserver is Karaf.
- 8. De beheerinterface wordt gemaakt in Vaadin.
- 9. De VDB-naar-SPOT-export schrijft de resultaten van de query's op de VDB-database weg in files voordat er wordt begonnen aan een update van de SPOT-database.
- 10. Waardes, die de werking van een bundle beïnvloeden, worden in een configuratiefile geschreven.
- 11. Ten behoeve van BI worden de KV5, KV6, KV15 en de KV17 berichten opgeslagen in aparte tabellen, één tabel per berichttype.
- 12. SIRI berichten worden in de MongoDB weggeschreven.
- 13. De PPT voldoet aan de TMI-PPT standaard, waarbij is gekozen voor de variant 'Basisprijzen in een directe prijs matrix'.
- 14. Voorheen was het zo dat CROW-NDOV (voorheen GOVI) gegevens kreeg DRIS-M (ATS). Op 22-07-2018 is besloten om deze gegevens indirect vanuit SPOT te leveren.



Gegenereerde code

4.1.3 XSD & WSDL

SPOT maakt gebruik van **gegenereerde code** voor de afhandeling van berichten (KV5, KV6, KV17 en KV15). De code wordt gegenereerd aan de hand van xsd-s en wsdl-s.

De door SPOT gebruikte xsd- en wsdl-bestanden worden gebundeld in een zip-bestand geleverd.

Dit betreft de volgende bestanden:

<i>Keten</i>	<i>Stroom</i>	<i>Bestand</i>
EBS	KV5	Kv5EBSService.wsdl
		Ebs_contracts.xsd
		Ebs_telegrams.xsd
	KV6	Kv6EBSService.wsdl
		Ebs_contracts.xsd
		Ebs_telegrams.xsd
	KV17	kv17EBSService.wsdl
		generic-ebs-v11.xsd
		kv17-ebs-v11.xsd

Illustratie context vertaalslag



Titel : Database Design Description (DBDD) SPOT
Kenmerk : SPOT-DBDD Versie : 2.1
Status : Voorlopig

QUAY (<u>NeTEx</u>)	<i>A place such as platform, stance, or quayside where passengers have access to PT vehicles, Taxi, cars or other means of transportation. A QUAY may serve one or more VEHICLE STOPPING PLACES and be associated with one or more STOP POINTS. A QUAY may contain other sub QUAYs. A child QUAY must be physically contained within its parent QUAY.</i>
QUAY (BISON-FHT)	<i>Een QUAY is iedere plaats, zoals een perron, een kade, een steiger, waar het in- en/of uitstappen daadwerkelijk plaatsvindt. Iedere plaats waar een voertuig direct langs halteert is een QUAY en een QUAY is altijd een plaats waaraan een voertuig halteert of kan halteren.</i>

BISON; libraries?

```
spot/SPOT/spot-adapter-kv17verrijking/src/main/java/nl/gvb/spot/adapter/kv17verrijking/KV17t
import org.apache.camel.Exchange ;
import org.apache.camel.Processor ;

import nl.connekt.bison.tmi8.kv17.msg.CHANGEDESTINATIONType ;
import nl.connekt.bison.tmi8.kv17.msg.CHANGEPASSTIMESType ;
import nl.connekt.bison.tmi8.kv17.msg.KV17CvlinfoType ;
import nl.connekt.bison.tmi8.kv17.msg.KV17JourneyType ;
import nl.connekt.bison.tmi8.kv17.msg.KV17MutatejourneyType ;
import nl.connekt.bison.tmi8.kv17.msg.KV17MutatejourneystopType ;
import nl.connekt.bison.tmi8.kv17.msg.LAGType ;
import nl.connekt.bison.tmi8.kv17.msg.MUTATIONMESSAGETYPE ;
import nl.connekt.bison.tmi8.kv17.msg.SHORTENType ;
import nl.gvb.spot.entity.Kv17 ;
import nl.gvb.spot.utils.XMLGC ;

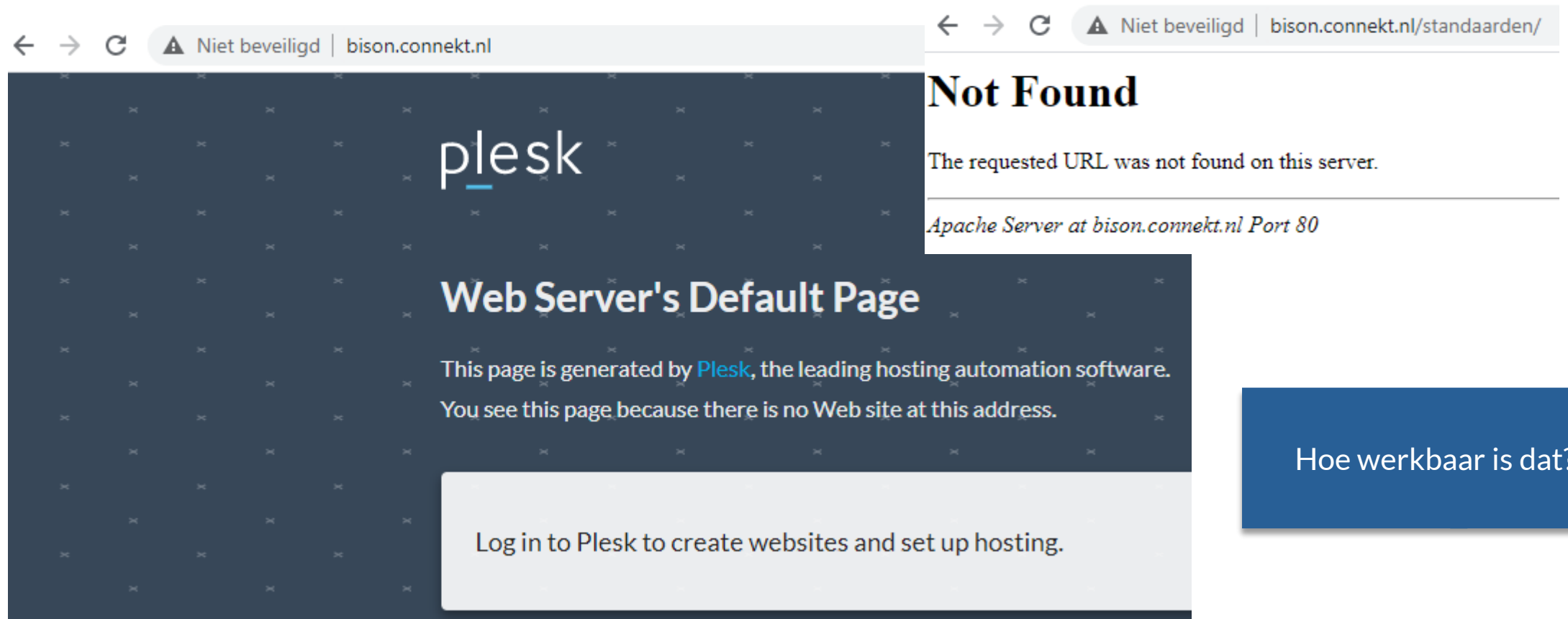
77 private Kv17 findKV17Type ( Serializable item , Kv17 kv17 , OffsetT
78     Kv17 result = null ;
79     if ( item instanceof CHANGEDESTINATIONType ) {
80         result = createChangeDestination ( item , kv17 ) ;
81     } else if ( item instanceof CHANGEPASSTIMESType ) {
82         result = createChangePasstimes ( item , kv17 , operatingDa
83     } else if ( item instanceof LAGType ) {
84         result = createLag ( item , kv17 ) ;
85     } else if ( item instanceof MUTATIONMESSAGETYPE ) {
86         result = createMutationMessage ( item , kv17 ) ;
87     } else if ( item instanceof SHORTENType ) {
88         result = createShorten ( item , kv17 ) ;
89     }
90     return result ;
91 }

93 private Kv17 fillRemainingFields ( Kv17 result , KV17JourneyType a , KV17MutatejourneyType b ) {
94     result.setDataownercode ( a.getDataownercode ( ) ) ;
95     result.setJourneynumber ( a.getJourneynumber ( ) ) ;
96     result.setLineplanningnumber ( a.getLineplanningnumber ( ) ) ;
97     result.setOperatingday ( XMLGC.toOffsetDateTime ( a.getOperatingday ( ) ) ) ;
98     result.setReinforcementnumber ( a.getReinforcementnumber ( ) ) ;

100     if ( b != null ) {
101         if ( b.getCANCEL ( ) != null ) {
```



Bereikbaarheid Bison



Hoe werkbaar is dat?



Duplicated code

spot/SPOT/spot-adapter-kvx-receivemessages/src/main/java/nl/gvb/spot/adapter/kvx/receivemessages/EBStoKV17.java

```

44 class EBStoKV17 implements Processor {
45     private static final int MINIMAL_YEAR = 2000 ;
46     private static final int SECONDS = 60 ;

48     private final SimpleDateFormat sdf = new SimpleDateFormat ( "HH:mm:ss" ) ;

50     private VDBPersist persister = new VDBPersistImpl ( ) ;

52     @ Override
53     public void process ( Exchange exchange ) {
54         Messages body = exchange . getIn ( ) . getBody ( Message
55         VVTMPUSH vvtmpush = new VVTMPUSH ( ) ;
56         List < KV17CvlinfoType > kv17s = new ArrayList < > ( )
57         if ( body != null && ! body . getMessage ( ) . isEmpty (
58             HeaderType header = body . getHeader ( ) ;
59             String dataowner = "GVB" ;
60             if ( header != null && ! "EBS" . equals ( header . g
61                 dataowner = header . getIssuer ( ) ;
62     }

```

references

- EBStoKV17.java (PATTERN_DUPLICATE_START)
44 spot/SIRI/siri-plugin-kvx-receivemessages/src/main/java/nl
- EBStoKV17.java (PATTERN_DUPLICATE_START)
44 spot/SPOT/spot-adapter-kvx-receivemessages/src/main/ja
- EBStoKV17.java (DUPLICATE_START)
44 spot/SIRI/siri-plugin-kvx-receivemessages/src/main/java/nl
- EBStoKV17.java (DUPLICATE_START)
44 spot/SPOT/spot-adapter-kvx-receivemessages/src/main/ja



Illustratie veelvoorkomend patroon overzetten data uit attributes

```
spot/SPOT/spot-adapter-pub-in/src/main/java/nl/gvb/spot/adapter/pub/ProcessPeriodes.java
160     pptPeriodes.setEinddatum ( periode.getEinddatum ( ) );
161     pptPeriodes.setLastindex ( periodes.getLastindex ( ) );
162     periodes.getPeriode ( ) .forEach ( periode -> pptPeriodes.getPeriode ( ) .add ( makePptPeriode ( periode ) ) );
163     return pptPeriodes ;
164 }

166 private nl.gvb.spot.adapter.pub.objects.pub_out.Periode makePptPeriode ( Periode p ) {
167     nl.gvb.spot.adapter.pub.objects.pub_out.Periode pptPeriode =
168     new nl.gvb.spot.adapter.pub.objects.pub_out.Periode ( ) ;
169     pptPeriode.setDataownercode ( p.getDataownercode ( ) );
170     pptPeriode.setEinddatum ( p.getEinddatum ( ) );
171     pptPeriode.setIndex ( p.getIndex ( ) );
172     pptPeriode.setIsbaseline ( p.isIsbaseline ( ) );
173     pptPeriode.setIsgepubliceerd ( p.isIsgepubliceerd ( ) );
174     pptPeriode.setKey ( p.getKey ( ) );
175     nl.gvb.spot.adapter.pub.objects.pub_out.Lijnen pptLijnen = new nl.gvb.spot.adapter.pub.objects.pub_
176     p.getLijnen ( ) .getLijn ( ) .forEach ( pptLijnen.getLijn ( ) :: add ) ;
177     pptPeriode.setLijnen ( pptLijnen ) ;
178     pptPeriode.setPublicatiedatum ( p.getPublicatiedatum ( ) );
179     pptPeriode.setStartdatum ( p.getStartdatum ( ) );
180     pptPeriode.setVersie ( p.getVersie ( ) );
181     pptPeriode.setZipfile ( p.getZipfile ( ) );
182     return pptPeriode ;
183 }

185 private File generateMetadata ( Periode periode , EnumKoppelvlakCode code ) {
186     File file = new File ( kv1IndexLocallocation
187     + periode.getZipfile ( ) .replace ( ".zip" , METADATA_EXTENSION ) .replace ( "KV1" , code.value ( ) ) );
188     OperatorFileMetadata mData = new OperatorFileMetadata ( ) ;
189     mData.setGeldigVanaf ( periode.getStartdatum ( ) );
190     mData.setGeldigTot ( periode.getEinddatum ( ) );
```



Illustratie logging middels eigen wrapper SpotLog

spot/SIRI/siri-plugin-siript-producer/src/main/java/nl/gvb/siri/plugin/producer/pt/RetrievePTFromDB.java

```
*/
23     final class RetrievePTFromDB implements RetrieveSnapshotFromDB < NotifyProductionTimetablePackage > {

25         private static final SpotLog LOG = new SpotLog ( "RetrievePTFromDB" );
26         private final PTPersister ptPersister = new PTPersisterImpl ( );

28         @Override
29         public NotifyProductionTimetablePackage getSnapshot ( List < Subscription > subscriptions ) {
30             LOG.info ( "Look voor valid PVJs" );

32             NotifyProductionTimetablePackage nntp =
33             new NotifyProductionTimetablePackage ( SiriID.generateProducerInfo ( "PT", null ),
34             new ProductionTimetableDeliveriesStructure ( ) );

36             LOG.info ( "RetrievePTFromDB::getSnapshot : " + ( nntp != null ? nntp.getNotification ( ) : "NULL" ) );

38             for ( Subscription subscription : subscriptions ) {
39                 LOG.info ( "RetrievePTFromDB::getSnapshot : subscription " + subscription.getAccountId ( ) );
40                 ProductionTimetableRequestStructure request =
41                 ( ProductionTimetableRequestStructure ) subscription.getRequest ( );
42                 nntp.getNotification ( ).getProductionTimetableDelivery ( )
43                 .add ( createPTR ( ptPersister.makeSnapshot ( request ), subscription ) );
44             }
45             nntp.getExtension ( ).getAny ( ).add ( ( Object ) JAXBUtil.createJAXBElement ( "snapshot", "snapshot" );
46             ptPersister.stop ( );
47             return nntp ;
48         }
    }
```



SIRI UK; library? gegenereerde code?

```
spot/SIRI/siri-plugin-request-consumer/src/main/java/nl/gvb/siri/plugin/consumer/request/HandleVehicleMonitoring.java

10     import nl.gvb.siri.utils.SiriID ;
11     import nl.gvb.spot.utils.XMLGC ;
12     import uk.org.siri.siri.VehicleMonitoringDeliveriesStructure ;
13     import uk.org.siri.siri.VehicleMonitoringDeliveryStructure ;
14     import uk.org.siri.wsd1.prod.WsServiceRequestInfoStructure ;

/**
 * Handle a Vehicle Monitoring request
 *
 * @author geerts/boogas
 */
21     class HandleVehicleMonitoring {
22         private final RequestConsumer requestConsumer ;
23         private final VMPersister vmPersister = new VMPersisterImpl ( "" ) ;

25         HandleVehicleMonitoring ( RequestConsumer requestConsumer ) {
26             this.requestConsumer = requestConsumer ;
27         }

29         void notify ( VehicleMonitoringPackage vmp ) {
30             WsServiceRequestInfoStructure info = vmp.getServiceRequestInfoStructure ( ) ;
31             VehicleMonitoringDeliveriesStructure response = new VehicleMonitoringDeliveriesStructure ( ) ;
32             if ( requestConsumer.getSiriAccountPersister ( ).verify ( info.getAccountID ( ), info.getAccountID ( ) ) ) {
33                 info.getRequestorRef ( ), info.getAddress ( ) ) {
```



spot-master bevat versies gebruikte libraries; wijkt af van documentatie

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>nl.gvb.spot</groupId>
  <artifactId>spot-master</artifactId>
  <version>4.0.1-SNAPSHOT</version>
  <name>MASTER SPOT (${maven.build.timestamp}) Project</name>
  <packaging>pom</packaging>

  <modules>
    <module>SPOT</module>
    <module>SIRI</module>
  </modules>

  <activemq.version>5.15.9</activemq.version>
  <build-helper-maven-plugin.version>3.0.0</build-helper-maven-plugin.version>
  <camel.version>2.24.2</camel.version>
  <cxfr.version>3.3.5</cxfr.version>
  <cxfr-codegen-plugin.version>3.3.3</cxfr-codegen-plugin.version>
  <h2.version>1.4.192</h2.version>
  <hamcrest.version>1.3</hamcrest.version>
  <hibernate-jpa-2.1-api.version>1.0.0.Final</hibernate-jpa-2.1-api.version>
  <hibernate.version>5.4.8.Final</hibernate.version>
  <!-- <hibernate.version>5.4.2.Final</hibernate.version> -->
  <hibernate-validator.version>6.1.0.Final</hibernate-validator.version>
  <httpclient-osgi.version>4.5.3</httpclient-osgi.version>
  <httpcore-osgi.version>4.4.6</httpcore-osgi.version>
```

Opdrachtgever	GVB
InTraffic kenmerk	SPOT SDD versie 2.3

Kenmerk	SPOT-SDD-2.3
Versie	V2.3
Datum	24-07-2019
Bestandsnaam	SDD SPOT V2.3_concept
Status	Concept

9.4 Bibliotheken

- Apache Camel 2.15.2
- Hibernate 4.3.6.Final
- Apache ActiveMQ 5.11.1
- OSGi 5.0.0

Inleiding SDD

1 Doel SDD

1.1 Identificatie

Dit is het Software Design Document van SPOT 1.0. Het document beschrijft de algemene opzet van SPOT:

- Uit welke onderdelen bestaat SPOT;
- Hoe werken deze onderdelen samen.

SPOT is een acroniem en staat voor “Single Point of Truth”.

SPOT is ontwikkeld om de kwaliteit van de Reisinformatievoorziening van het GVB te verbeteren.



Handleiding Operationeel Beheer; fatsoenlijke inleiding

Opdrachtgever :	GVB
Auteur :	SPOT-team (<i>InTraffic</i>)
Kenmerk :	SPOT-HLOB-1.8
Versie :	1.8
Datum :	17-11-2020
Bestandsnaam :	HLOB SPOT v1.8
Status :	Definitief



3.1.4.5 Station bewust overslaan

Er wordt besloten dat een trein toch niet zal stoppen op een bepaald station.

Scenario	Details
ATS wijzigt de <u>planning</u> voor een rit.	Zie § 6.6.5 van IRS ATS-DRIS
	Bij deze rit geldt nu voor de <u>betreffende</u> call: • AimedArrival/DepartureTime identiek • Arrival/DepartureBoardingActivity=passthru
ATS stuurt SIRI-ET bericht voor de gewijzigde rit (binnen tijdsvenster).	Een <i>EstimatedTimetableDelivery</i> met voor de gewijzigde rit een <i>EstimatedVehicleJourney</i> .
SPOT stuurt SIRI-ET (gefilterd) door.	Dezelfde <i>EstimatedTimetableDelivery</i> .
ATS stuurt SIRI-VM bericht voor de gewijzigde rit.	Een <i>VehicleMonitoringDelivery</i> met voor de gewijzigde rit een <i>VehicleActivity</i> .
SPOT stuurt SIRI-VM (gefilterd) door.	Dezelfde <i>VehicleMonitoringDelivery</i> .
ATS stuurt SIRI-SM bericht voor het betreffende <u>metrostation</u> .	Voor dit <u>station</u> een <i>StopMonitoringDelivery</i> met voor de gewijzigde passage een <i>MonitoredStopVisit</i> .
SPOT stuurt voor de bijbehorende <u>metrohalte</u> een SIRI-SM bericht .	Voor deze <u>halte</u> een <i>StopMonitoringDelivery</i> met dezelfde gegevens als in het bericht van ATS.

3.1.4.6 Afwijkende route

Dat is nogal lastig bij de metro...

Rationale Karaf / OSGi

- SPOT kent meerdere datastromen, waarbij elke stroom onafhankelijk bediend kan worden. Ook is mogelijk om een specifieke stroom te updaten. Omdat het hier om real-time informatie gaat, is het belangrijk dat de rest van de stromen geen last ondervinden van deze activiteiten. OSGi (Open Services Gateway initiative) is een standaard voor Java die is ontwikkeld om een modulaire systeem op te zetten, waarin precies deze handelingen (stoppen, starten, updaten) mogelijk zijn. Dit omdat per stroom een proces aangemaakt wordt, dat onafhankelijk van de andere processen gestopt, gestart en geüpdatet kan worden.

Wordt dat in de praktijk ook zo gebruikt?

Zijn er library conflicten?



OSGi Activator

spot/SPOT/spot-adapter-kvx-receivemessages/src/main/java/nl/gvb/spot/adapter/kvx/receivemessages/KVxActivator.java

```
1 package nl.gvb.spot.adapter.kvx.receivemessages ;

3 import org.osgi.framework.BundleActivator ;
4 import org.osgi.framework.BundleContext ;

/**
 * Start or stop the KVxReceiveMessages.
 *
 * @author geerts
 */
11 public class KVxActivator implements BundleActivator {
12     private KVxReceiveMessages kvx = new KVxReceiveMessages ( ) ;

14     @Override
15     public void start ( BundleContext context ) {
16         kvx.init ( ) ;
17     }

19     @Override
20     public void stop ( BundleContext context ) {
21         kvx.stop ( ) ;
22     }
23 }
```

spot/SIRI/siri-utils/src/main/java/nl/gvb/siri/Utils/BaseActivator

```
1 package nl.gvb.siri.Utils ;

3 import org.osgi.framework.BundleActivator ;
4 import org.osgi.framework.BundleContext ;
5 import org.osgi.framework.BundleException ;
6 import org.osgi.util.tracker.ServiceTracker ;

8 import nl.gvb.siri.interfaces.Plugin ;
9 import nl.gvb.siri.interfaces.SiriBase ;
10 import nl.gvb.spot.Utils.SafeRun ;
11 import nl.gvb.spot.Utils.SpotLog ;

/**
 * Start and stop a Plugin, and connect it to the SiriBase, by looking up the SiriBase serv
 * Lost, the Plugin is stopped.
 *
 * @author geerts
 */
19 public abstract class BaseActivator implements BundleActivator {
20     private static final SpotLog LOG = new SpotLog ( "BaseActivator" ) ;
21     private Plugin plugin ;
22     private ServiceTracker < SiriBase , SiriBase > st ;
23     private BundleContext context ;
24     private boolean stopping = false ;

26     public BaseActivator ( Plugin plugin ) {
27         this.plugin = plugin ;
28     }
```

Zowel super class
als directe
implementaties
gebruikt



GUI / Inversion of Control

spot/SPOT/spot-web/src/main/java/nl/gvb/spot/web/views/KV15View.java

```
16 import nl.gvb.spot.entity.Kv15 ;
17 import nl.gvb.spot.entity.Stoppoint ;
18 import nl.gvb.spot.persistence.ui.KVxQuery ;
19 import nl.gvb.spot.persistence.ui.Kv15Persist ;
20 import nl.gvb.spot.persistence.ui.impl.Kv15PersistImpl ;
21 import nl.gvb.spot.persistence.vdb.VDBPersist ;
22 import nl.gvb.spot.persistence.vdb.impl.VDBPersistImpl ;
23 import nl.gvb.spot.web.BisonFields ;
24 import nl.gvb.spot.web.TextConstants ;
25 import nl.gvb.spot.web.enums.KV15MessageTypes ;

/**
 * View for KV15 messages
 */
30 public class KV15View extends MessagesView < Kv15 > {
31     private static final long serialVersionUID = -696163053997234221 L ;

32     private final Label lblNoMessagecodenumbersAvailable =
33         new Label ( TextConstants.none ( TextConstants.MESSAGECODENUMBER ) ) ;
34     private final Label lblNoUserstopcodeAvailable = new Label ( TextConstants.none ( TextConstants.USERSTOPCODE ) ) ;
35     private final CheckBox chboxMessagecodenumbers = new CheckBox ( TextConstants.MESSAGECODENUMBER ) ;
36     private final CheckBox chboxUserstopcodes = new CheckBox ( TextConstants.USERSTOPCODE ) ;
37     private final ComboBox cboxMessagecodenumbers = new ComboBox ( ) ;
38     private final ComboBox cboxUserstopcodes = new ComboBox ( ) ;
```

Rechtstreekse
afhankelijkheid op
implementatie i.p.v.
interface



BONCAT

Navigation

- siri-plugin-opengeo
 - src
 - main
 - java
 - nl
 - gvb
 - siri
 - plugin
 - opengeo
 - OpenGeoActivator.java
 - OpenGeoFileParser.java
 - SiriOpenGeoService.java
- + siri-plugin-receivemessages
- + siri-plugin-request-consumer
- + siri-plugin-riv-producer
- siri-plugin-sendmessages
 - src
 - main
 - java
 - nl
 - gvb
 - siri
 - plugin
 - sendmessages
 - AbstractSendMethods.java
 - SendActivator.java

Top 100 Function Size In LOC

- logToFile(NotifyPackage < ? > s , String fileLocation) in Handle
- getDetailedTable(Integer listID) in KV17View.java (48 lines)
- getDetailedTable(Integer listID) in KV6View.java (48 lines)
- writeToFile(String collectionname , LocalDate checkDate , Sort
- createForEachLineOrQuay(SubscribePackage sp , String fileName
- create(String routes) in MainCreator.java (41 lines)
- getDetailedTable(Integer listID) in KV15View.java (40 lines)
- createKV15(STOPMESSAGEType sm) in Kv15Builder.java (38
- baseMonitoredCallOnOnwardCall(AbstractMonitoredCallStr
- translateToEt(DatedVehicleJourneyStructure dvjPT , DatedTi
- sendStructure(final SiriProducerStructure s , final SiriProduce
- configure() in BaseRoutes.java (36 lines)
- getDetailedTable(Integer listID) in SiriETView.java (35 lines)
- updateTabpanel() in UsersView.java (34 lines)
- addFileToTable(Kv1 kv1 , String fileName , boolean isKv1 , int i

MAINBEST PRACTICESDEPENDENCY TABLEWORD CLOUD

DECOMPOSITION

very low risk

low risk

medium risk

high risk

very high risk

DECOMPOSITION

overall rating

dependency volume

dependency span

function complexity

function size

function parameter

VOLUME

duplicate tokens

tokens

DECOMPOSITION

Metric	Value
Weighted Decomposition Rating	59.34
Function Parameter Rating	97.76
Function Size Rating	37.36
Function Complexity Rating	88.79
Dependency Span Rating	50.85
Dependency Volume Rating	35.18

BEST PRACTICES

Empty Catches	0
Upper Level Catches	0
Illegal Catches	Not Tracked
File Aggregated Cyclic Dependency Weight	0
Unfinished Work	Not Tracked

VOLUME

Volume In Lines Of Code	279
Volume In Lines	329
Lines Of Comment	15
Volume In Tokens	2540
Duplicate Tokens Percentage	39.1%
Duplicate Pattern Tokens Percentage	51.2%
Churn In Lines	Not Tracked
Churn Percentage	Not Tracked

Planning

Next steps



Planning

1. Validatiesessie

Doelstelling:

- Draagvlak realiseren bij de betrokkenen voor de uitkomsten van het onderzoek.
- Presenteren en valideren bevindingen, onder andere aan de hand van voorbeelden.
- Gezamenlijk bepalen false positives.

Betrokkenen:

- Lead-architect / lead-ontwikkelaar(s) van InTraffic.

2. Rapportage

doelstelling:

- Beantwoorden onderzoeksvragen.
- Samenvatten bevindingen.
- Benoemen aanbevelingen incl. eventuele quick wins.

Betrokkenen:

- Opdrachtgever en andere stakeholders.

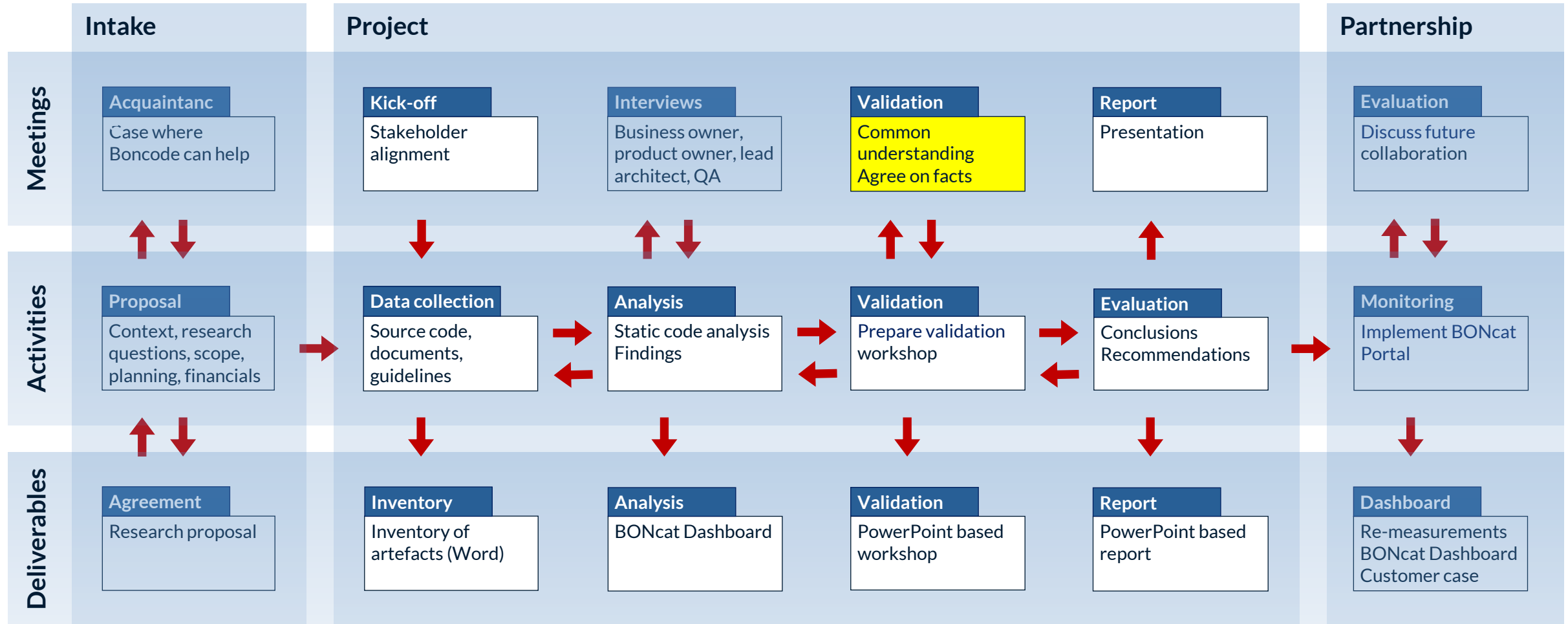


Planning (according to proposal)

Research	22/3– 26/3	29/3 – 2/4	5/4 – 9/4	12/4 – 16/4	19/4 - 23/4
Kick-off	24-3-2021				
Transfer of source code & documents	23-3-2021				
Analysis					
Validation			6-4-2021		
Analysis					
Final report					21-4-2021



Next steps



Questions, remarks?



J e r o e n M e e t s m a
j . m e e t s m a @ b o n c o d e . n l