

Managing an Industrial Software Rearchitecting Project with Source Code Labelling

Brice Govin, Nicolas Anquetil, Anne Etien, Stephane
Ducasse, Arnaud Monegier

Content

- Software Rearchitecting
- Rearchitecting project in Thales Air Systems
- Managing the rearchitecting Project
- Conclusion

Software Rearchitecting

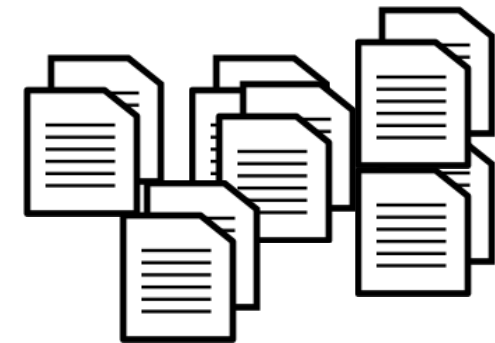
Legacy Software



Old



Valuable

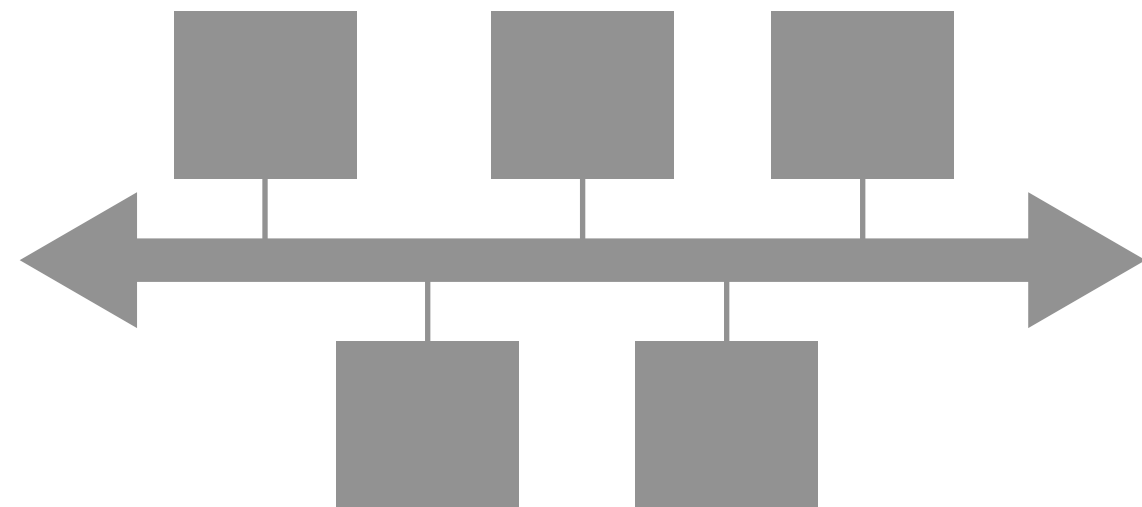
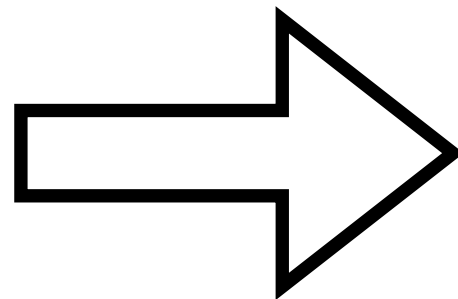
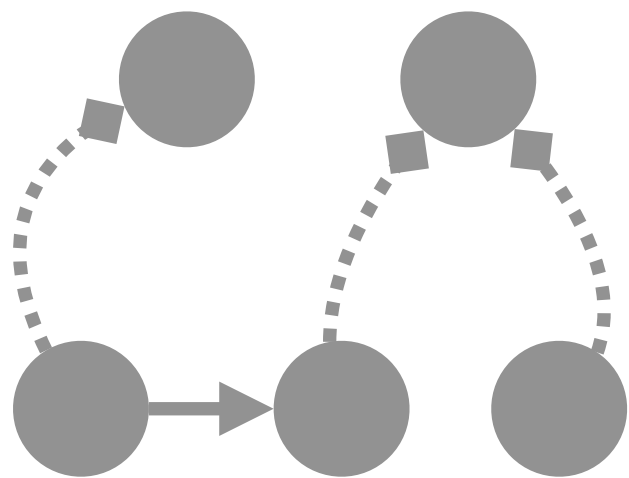


**Large
(often)**

Software Rearchitecting

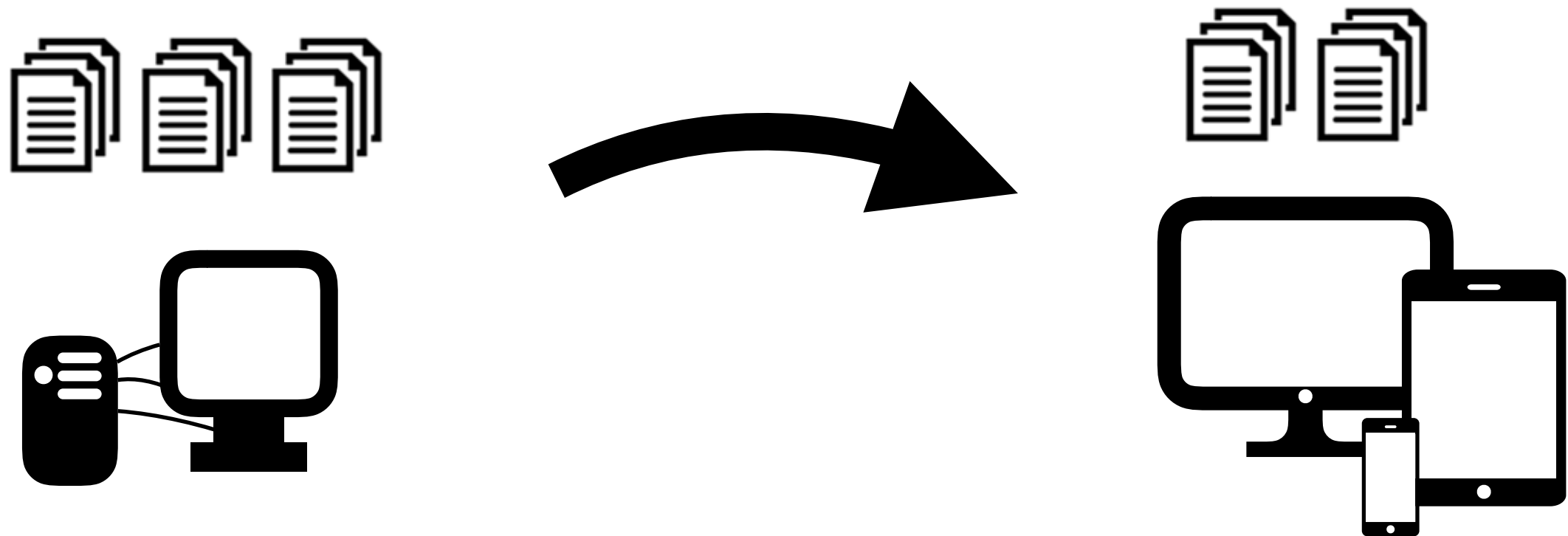
Rearchitecting

***“Software rearchitecting is
the process of changing software architecture”***

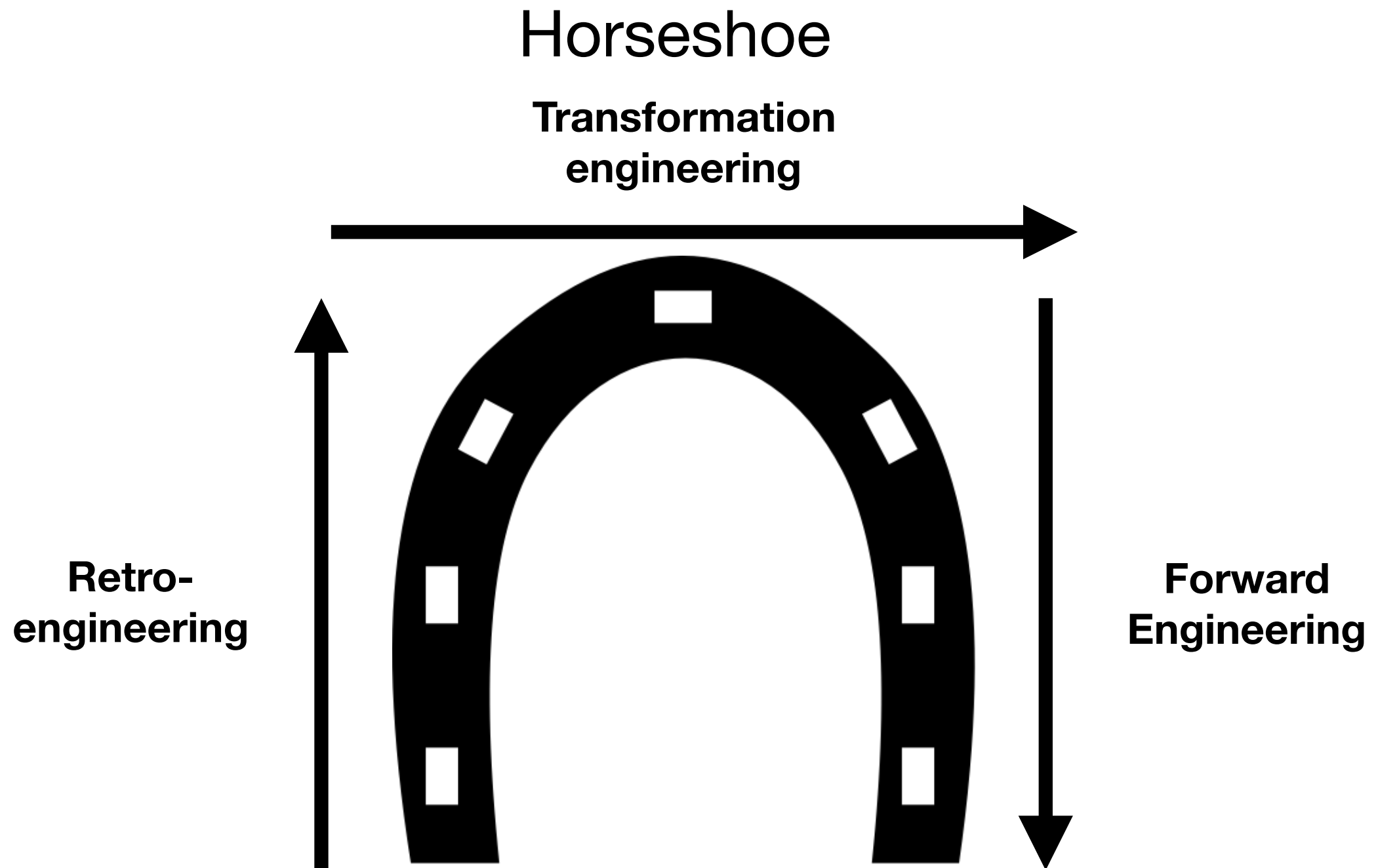


Software Rearchitecting

Thales goals

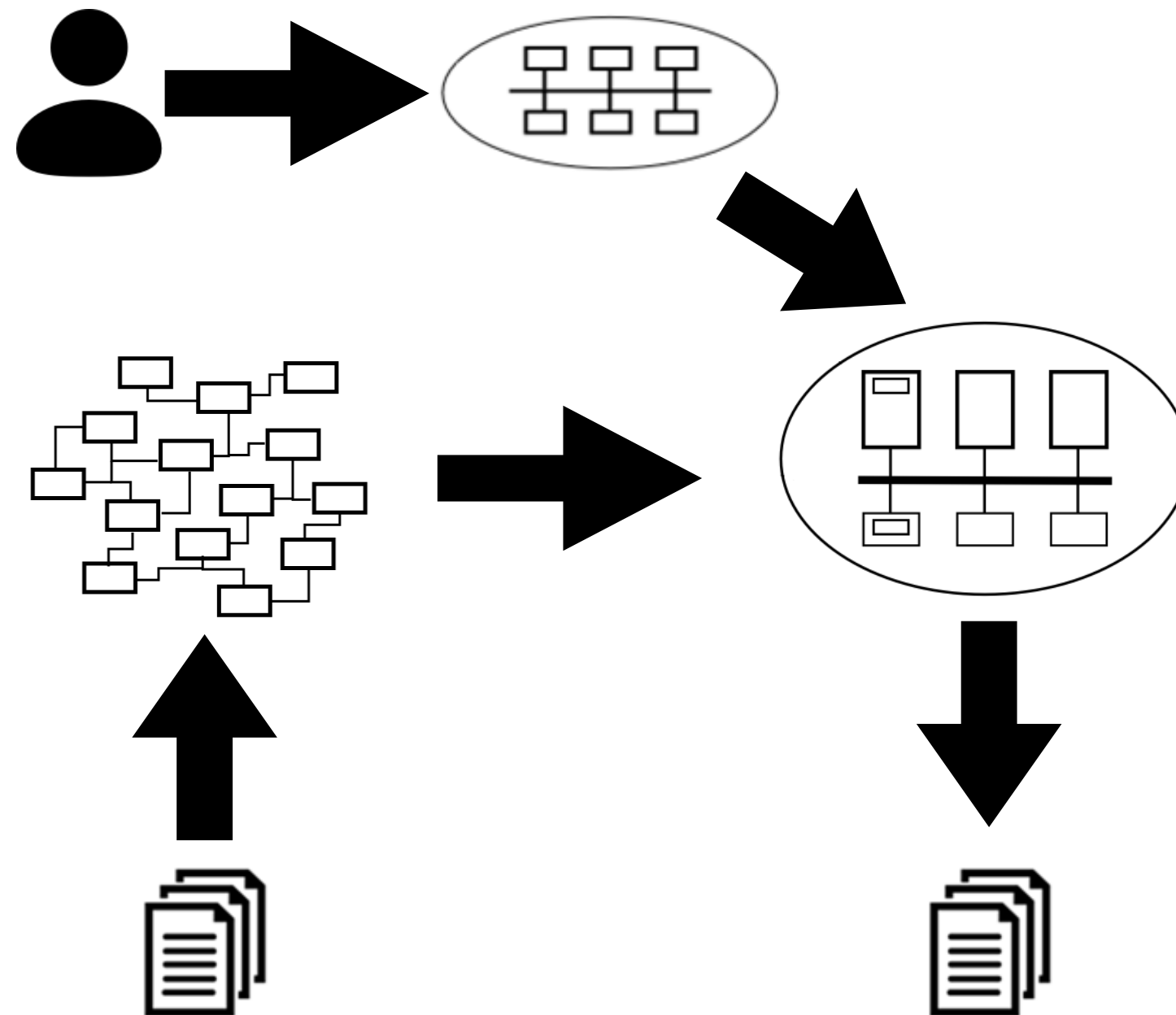


Project at Thales



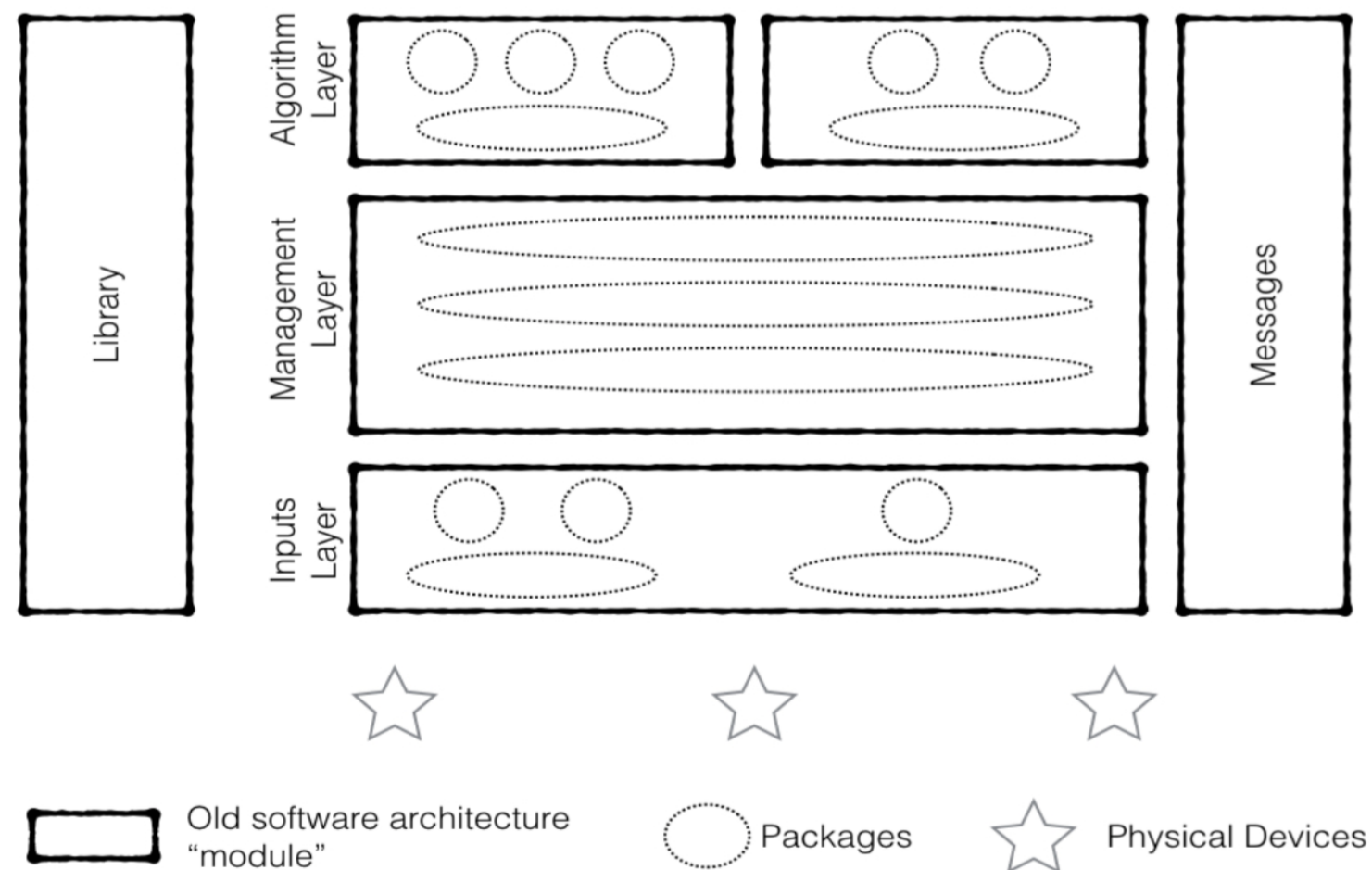
Project at Thales

Horseshoe in the project



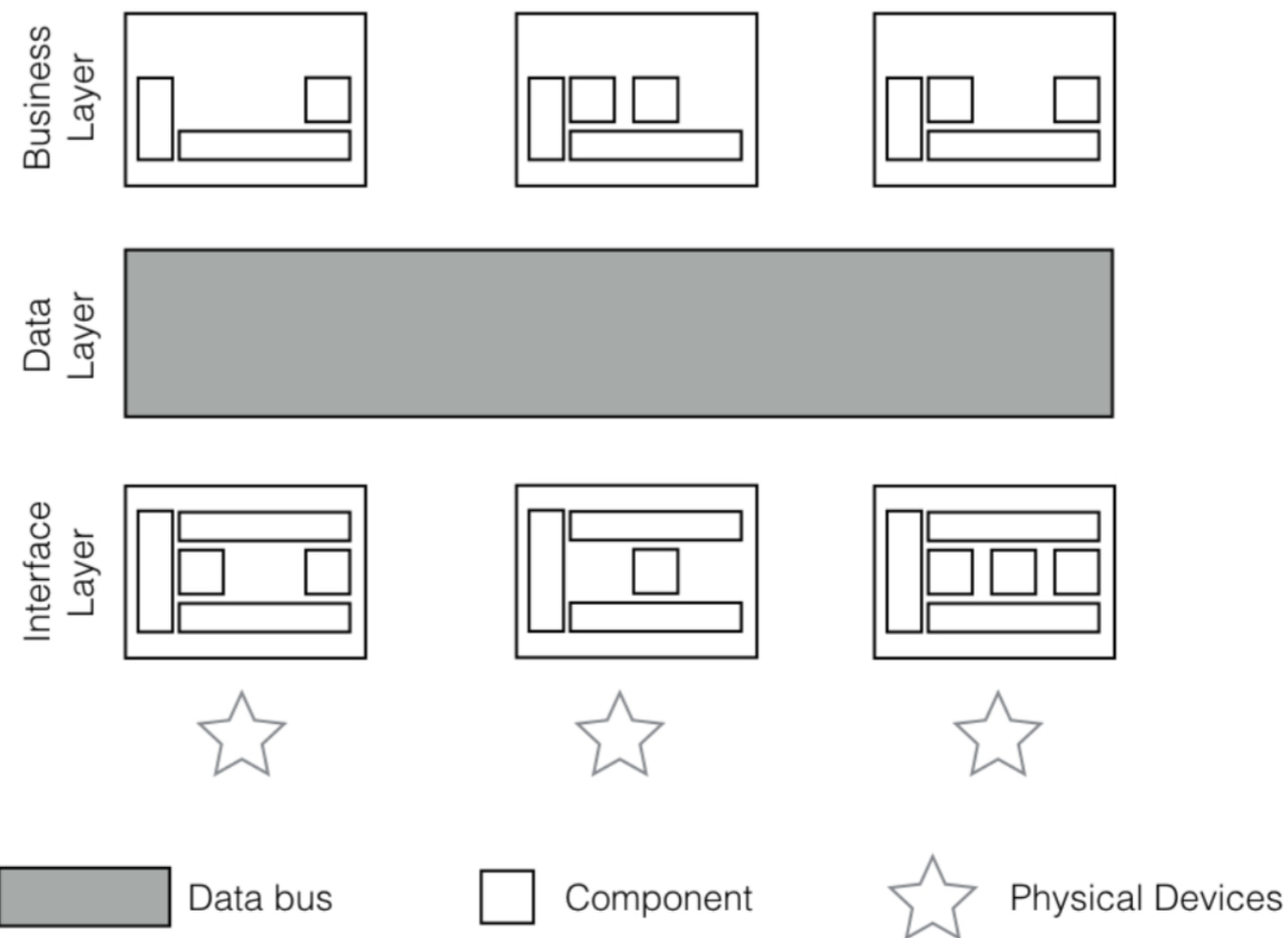
Project at Thales

Existing Architecture



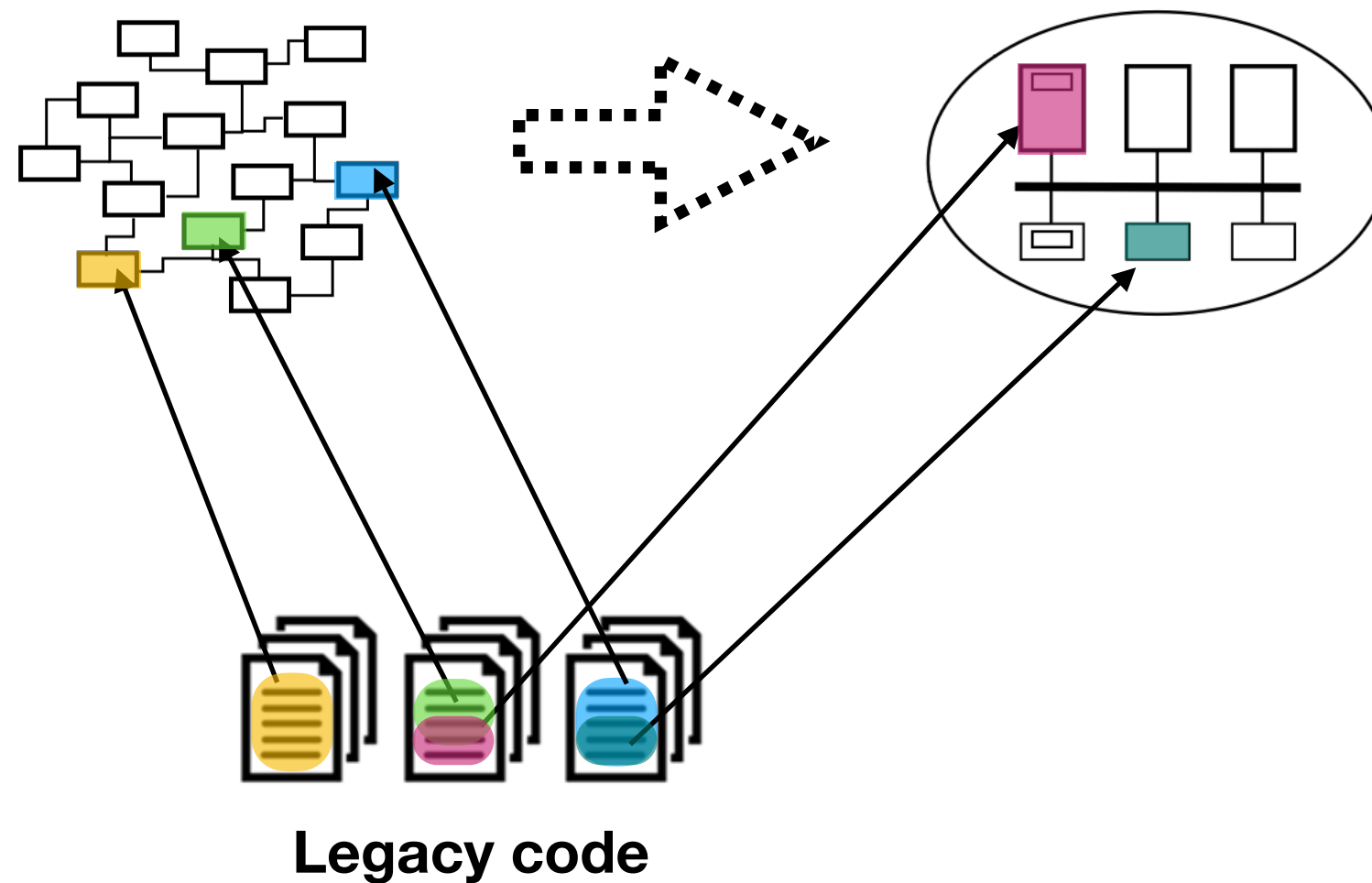
Project at Thales

New Architecture



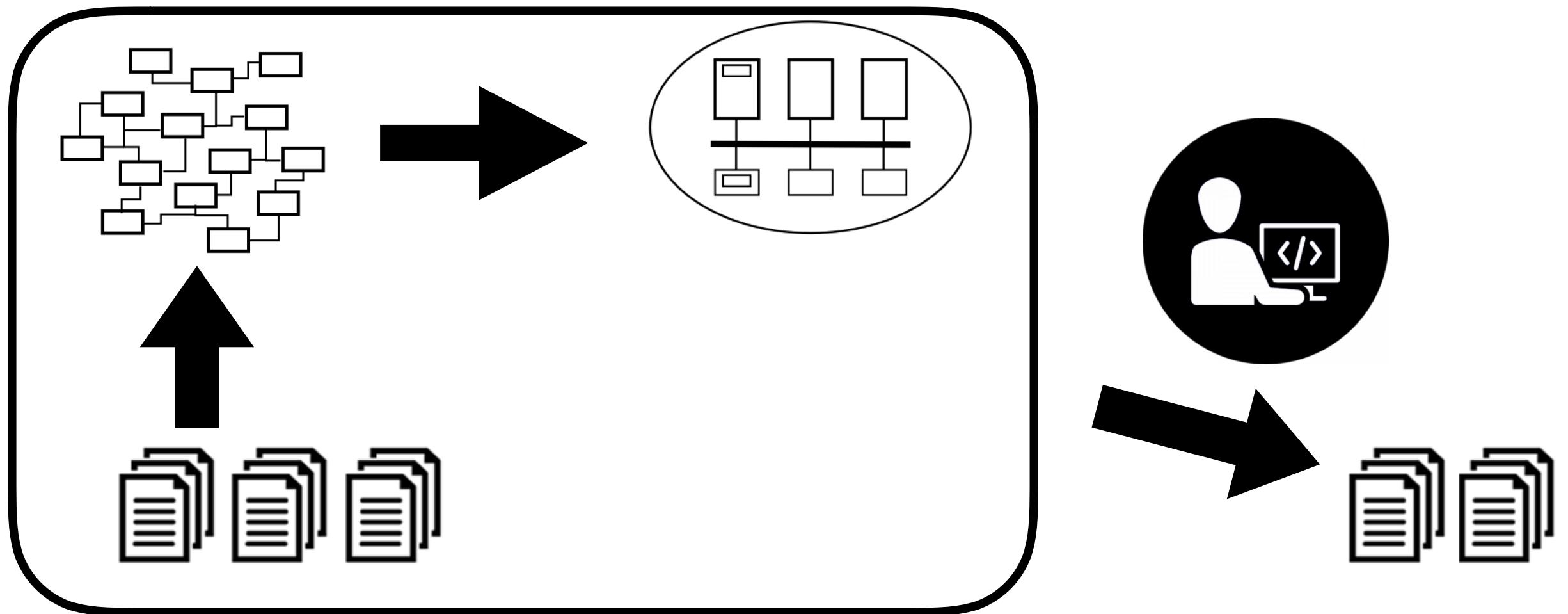
Project at Thales

Existing to New Architecture



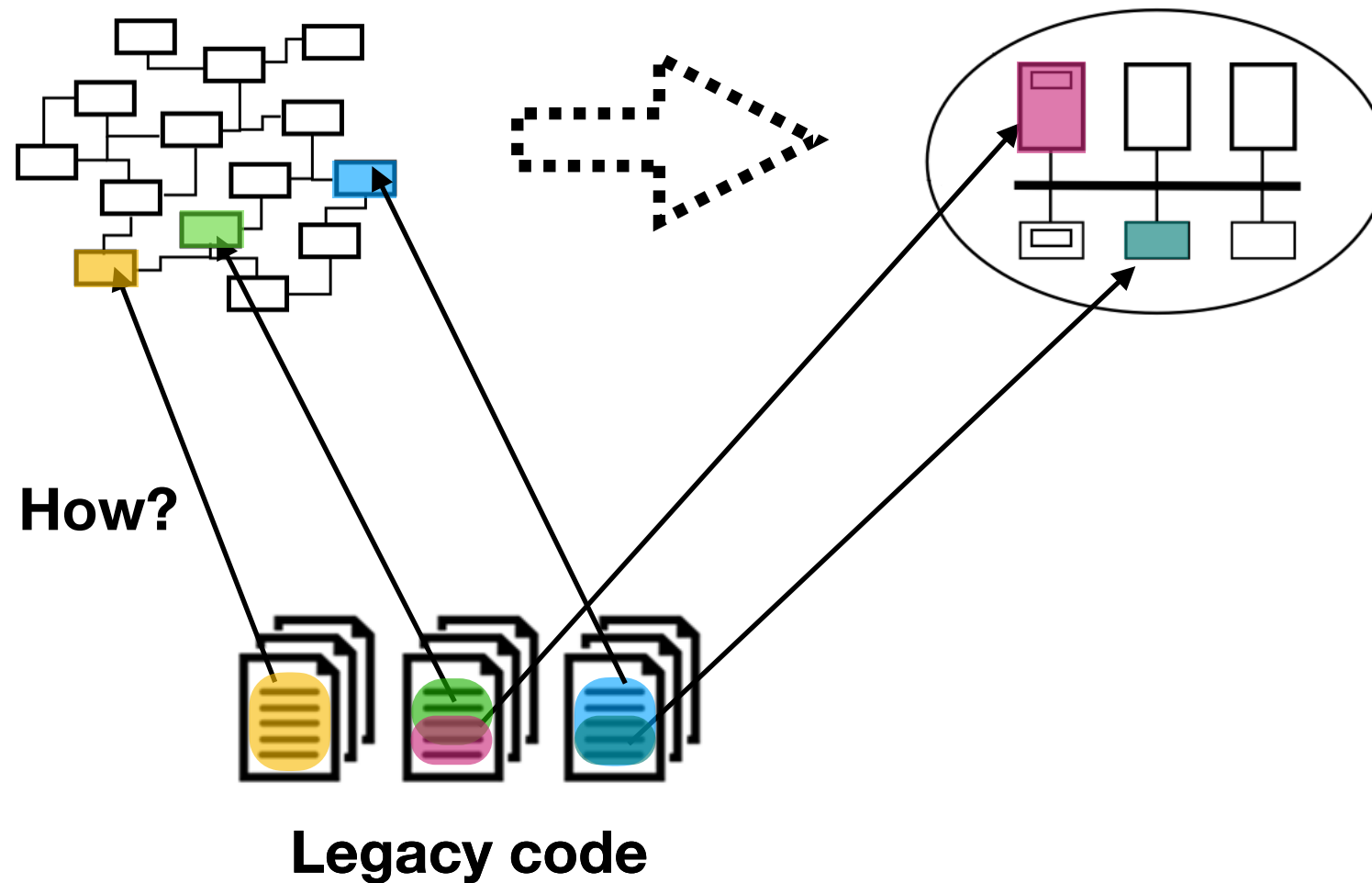
Project at Thales

Redevelopment



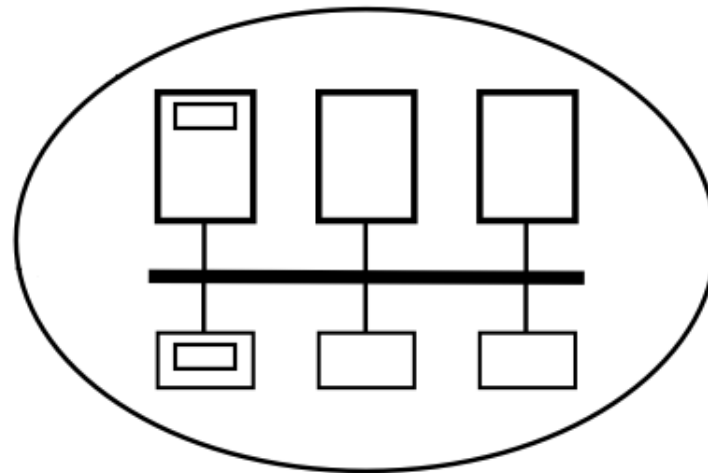
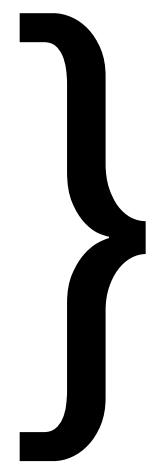
Project at Thales

Problem ?



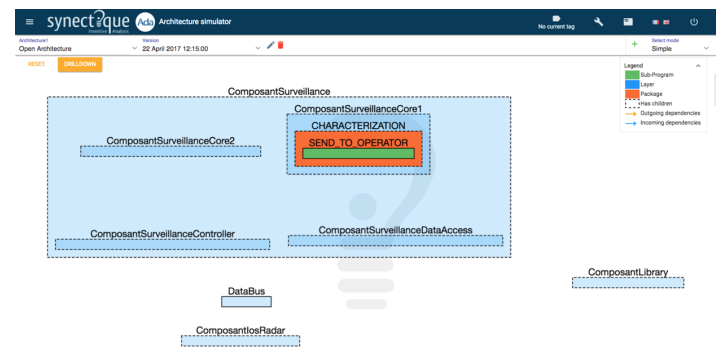
Managing the rearchitecting projet

Tag System

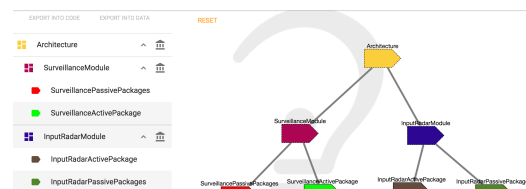


Managing the rearchitecting projet

Tags and application



Metric	Value
Tagged entities	
Number of tagged packages	8
Number of tagged subprograms	0
Navigation metrics	
Number of incoming links	12
Number of outgoing links	12
Data	
Number of data produced	30
Number of data consumed	33
Count	
Number of characters	15945
Number of lines	548
Number of lines of code	421



```

-- Code du message
MSG_CODE : constant COMMON_TYPES.T_UNSIGNED := 3;

-- Format applicatif
type T_ALARM_TYPE is
  (ERROR_1,
  ERROR_2,
  ERROR3);

type T_ALARM is
  record
    TYP : T_ALARM_TYPE;
    INVOLVED_TRACK_CODE : TRACKING_TYPES.T_TRACK_
  end record;

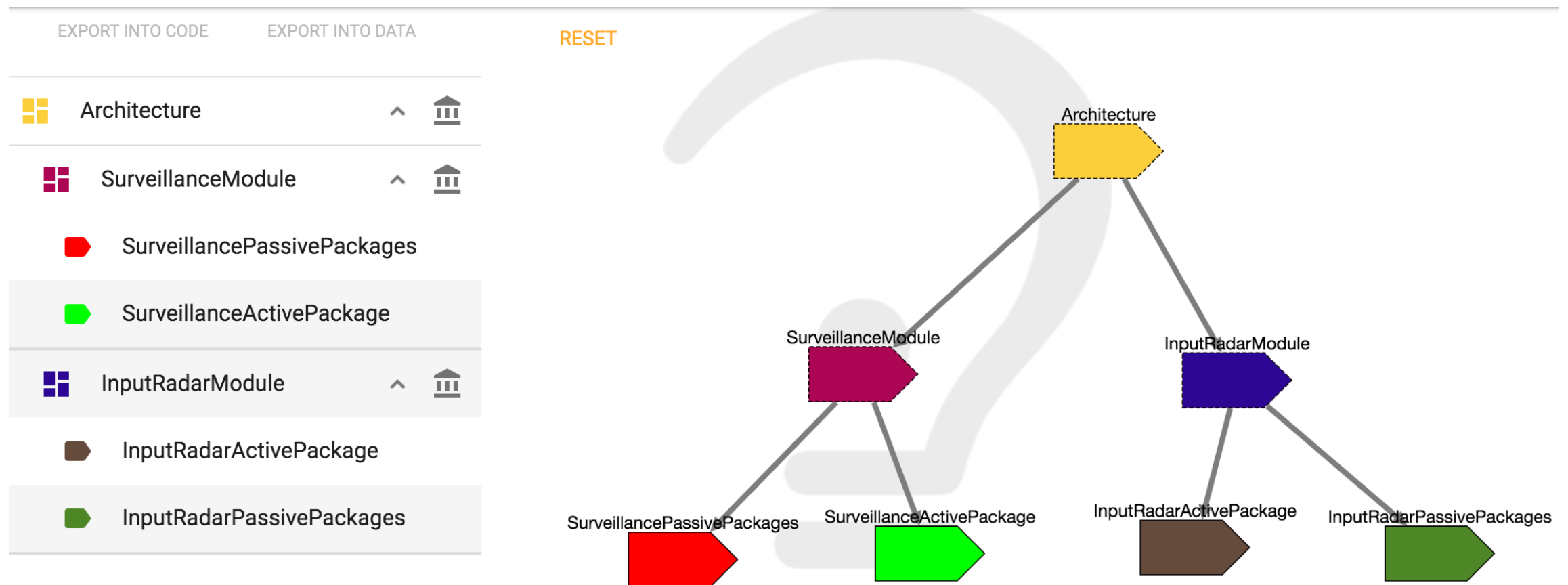
type T_ACCESS_ALARM is access T_ALARM;

-- Format ligne
MAX_SIZE : constant := 1;
subtype T_ID_ROUGH_ALARM is BASIC_DEFINITIONS.T_ID;
type T_ACCESS_ID_ROUGH_ALARM is access T_ID_ROUGH_A

function LINE_MSG MSG : in T_ALARM) return T_ID_ROU
  
```

Managing the rearchitecting projet

Materializing an architecture



Managing the rearchitecting projet

Allocating source code

The screenshot displays the synectique Ada Code Analyzer interface. The main window shows the source code for `cms_interface_9.ads` with the following content:

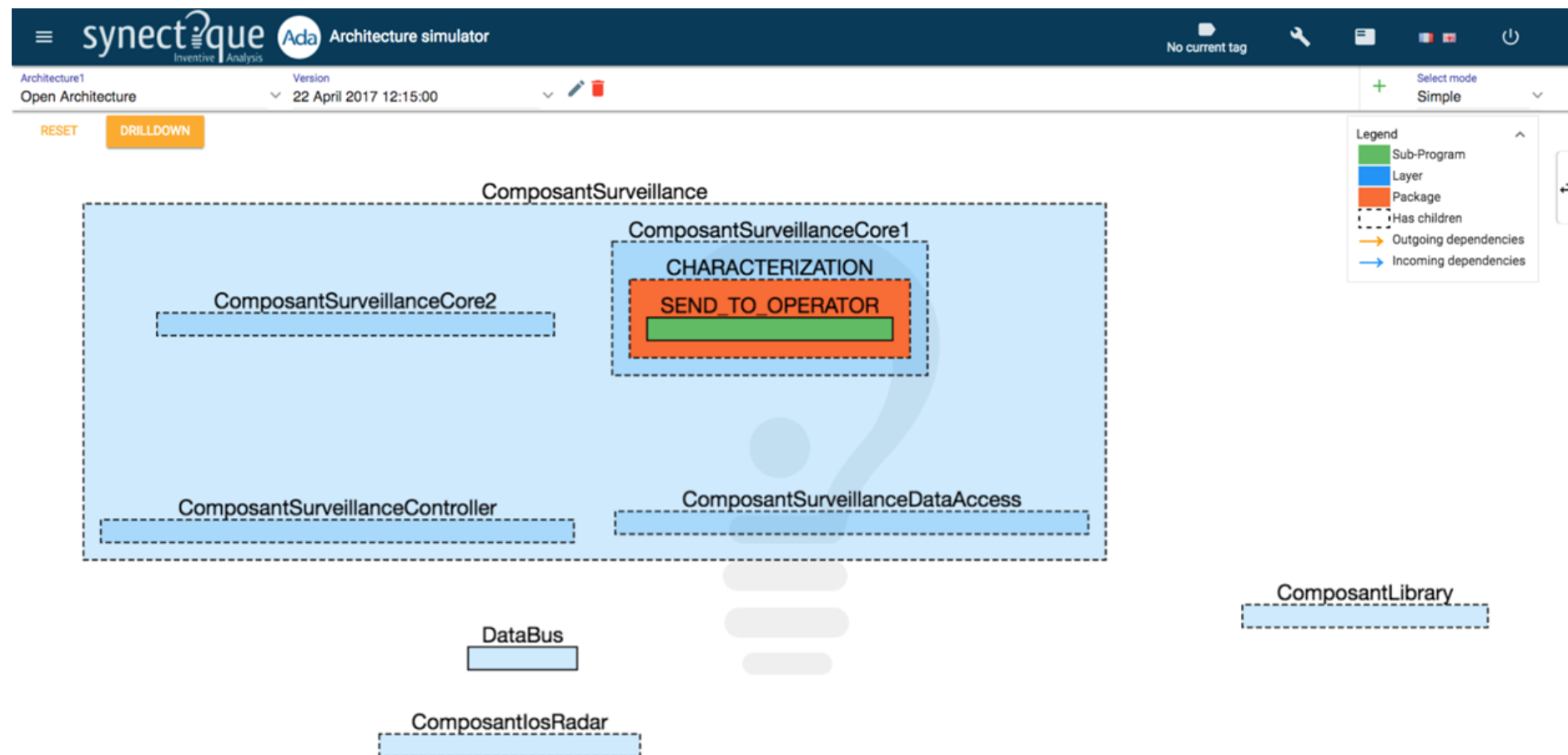
```
16
17 --! PURPOSE => Get the AI in order to send the generic action acknowledge
18
19 --!
20 with the right AI
21
22 -----
23 procedure Get_AI_for_transition_to_integrated(
24
25     AI : in out CS_Common_Structures.T_AI);
26
27 -- END FT4070439 (2)
28
29
30
31
32
33 --! FUNCTION => Init
34
35 --! PURPOSE => Initialise Action Module
36
37 -----
38
39 function Init
40
41 return Boolean;
42
43
44
45 -----
46
47 --! FUNCTION => Cleanup
48
49 --! PURPOSE => Activates shutdown procedures for package
50
51 -----
```

On the left, the 'ENTITIES MATCHING FILTERS' panel lists various entities like `CMS_Interface_9`, `CMS_Time`, `CSCI_Main`, etc. On the right, the 'ENTITIES DEPENDING ON ME' and 'ENTITIES I DEPEND ON' panels show dependencies. At the bottom right, a table displays code metrics:

Property	Value
Cyclomatic complexity	7
Number of children	10
Number of dead children	0
Number of external providers	4

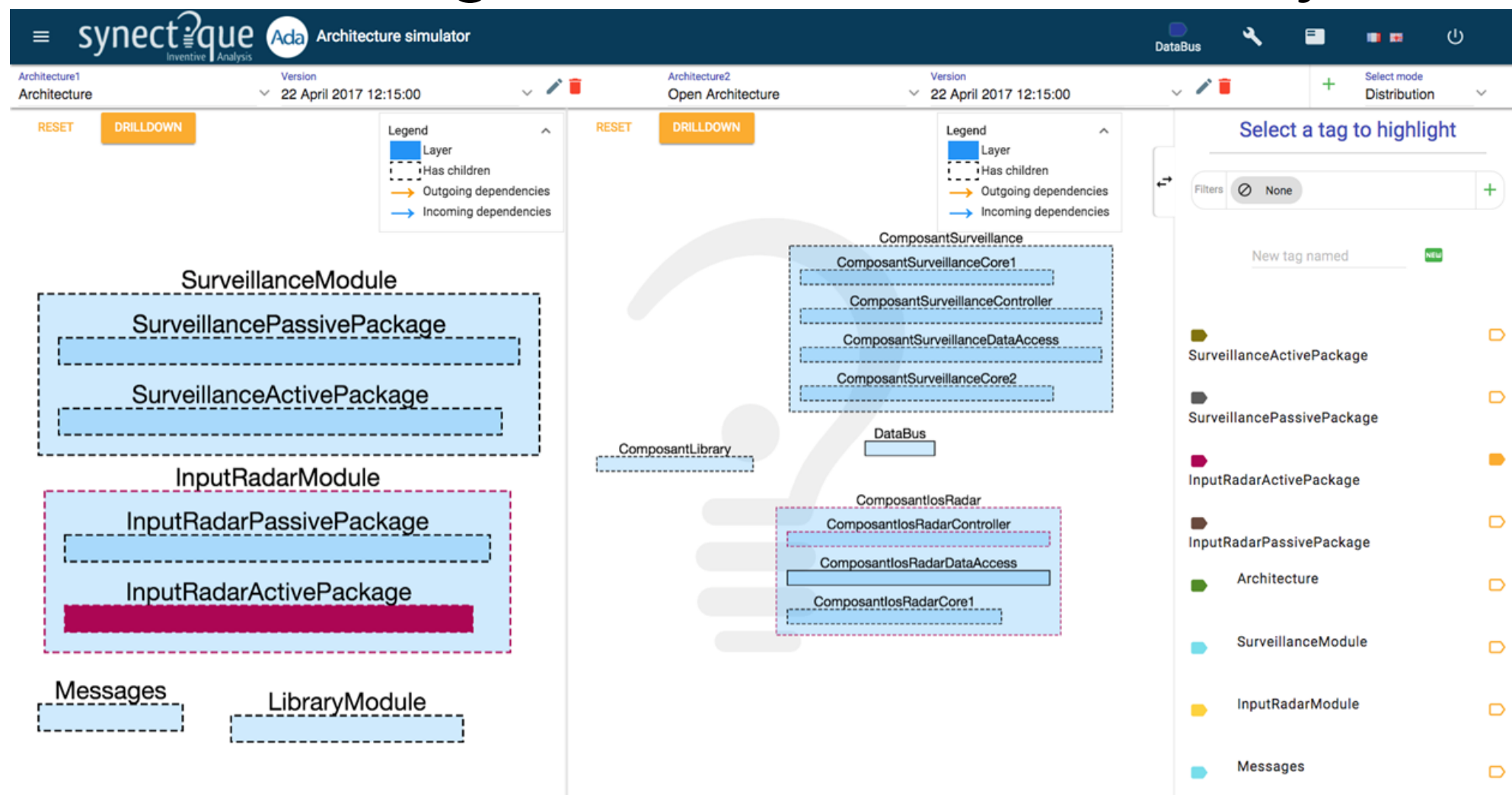
Managing the rearchitecting projet

Allocating source code



Managing the rearchitecting projet

Ensuring architecture consistency



Managing the rearchitecting projet

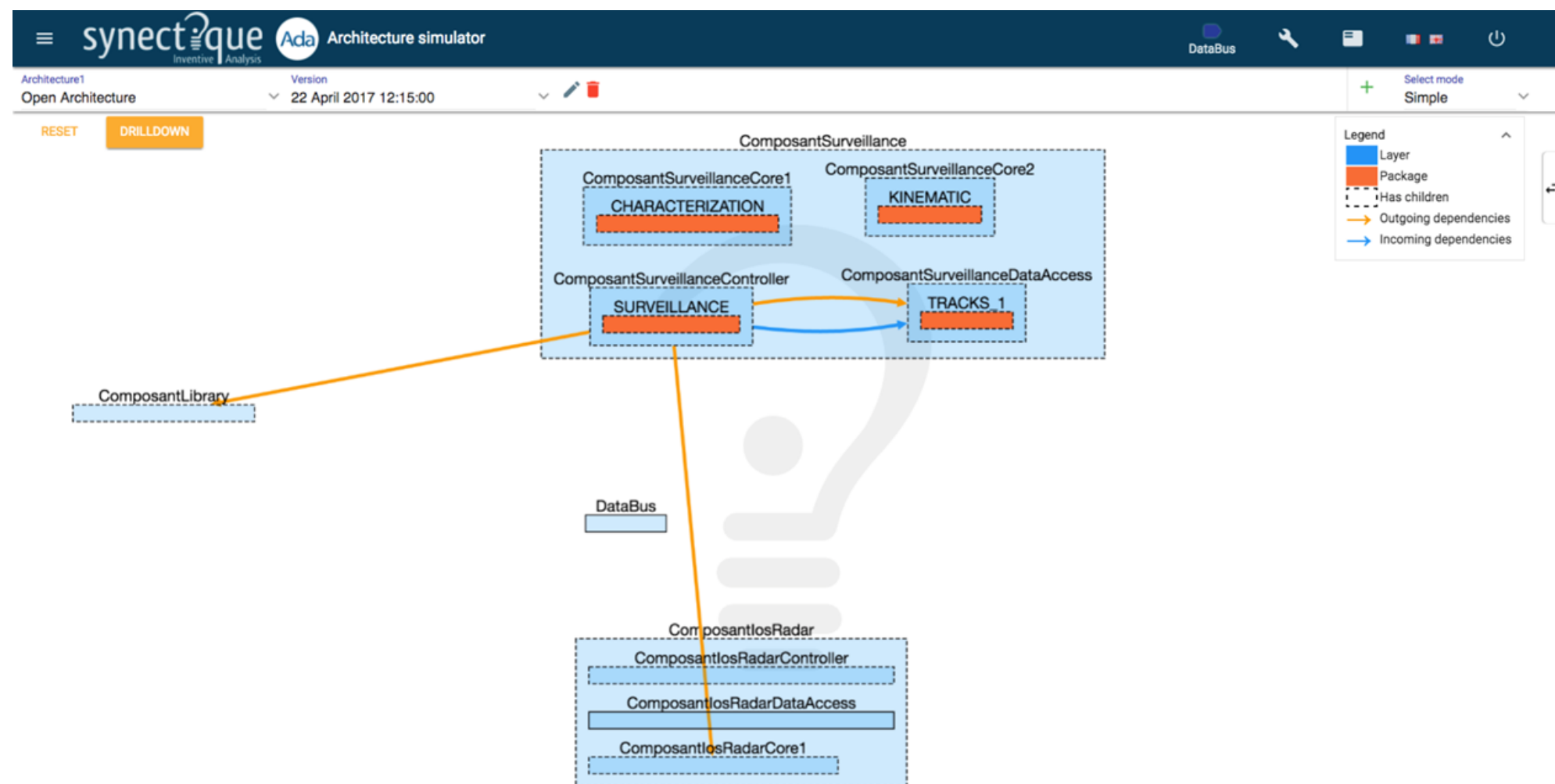
Ensuring architecture consistency

The screenshot displays the Invenio Analysis tool interface. On the left, a sidebar shows a tree view with 'Architecture' expanded, containing 'SurveillanceModule' and 'InputRadarModule'. The main area features a top toolbar with 'EDIT', 'DELETE', 'SPLIT', 'MERGE', and 'VISUALIZE' buttons. Below this is the 'Edit your tag' form, which includes a 'Categories' dropdown set to 'None', a 'Select parent tag' dropdown set to 'Architecture', a 'Tag name' field containing 'SurveillanceModule', a 'Description' field, and a 'Choose a color' button. A 'SAVE' button is at the bottom of the form. On the right, a 'METRICS' table is highlighted with a red border. The table has two columns: 'Metric' and 'Value'. It lists various metrics grouped into 'Tagged entities', 'Navigation metrics', 'Data', and 'LCount'.

Metric	Value
Tagged entities	
Number of tagged packages	8
Number of tagged subprograms	0
Navigation metrics	
Number of incoming Links	12
Number of outgoing links	12
Data	
Number of data produced	30
Number of data consumed	33
LCount	
Number of characters	15945
Number of lines	548
Number of lines of code	421

Managing the rearchitecting projet

Ensuring architecture consistency



Managing the rearchitecting projet

Locating architecture in source code

The screenshot displays the 'synect?que Ada Code Analyzer' interface. The main window shows the source code of 'cms_interface_9.ads' with line numbers 10 to 45. The code includes comments and a procedure definition:
--! PROCEDURE => Get_AI_for_transition_to_integrated
--! PURPOSE => Get the AI in order to send the generic action acknowledge
--! with the right AI
procedure Get_AI_for_transition_to_integrated(
AI : in out CS_Common_Structures.T_AI;
-- END FT4870439 (2)

--! FUNCTION => Init
--! PURPOSE => Initialise Action Module

function Init
return Boolean;

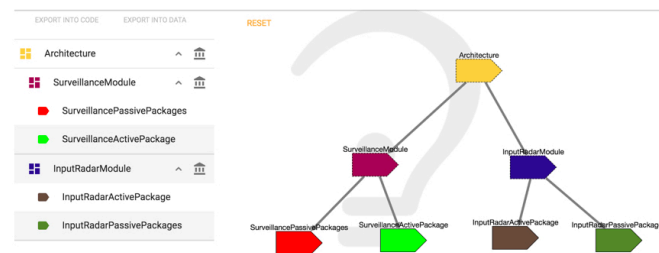
The interface also features a left sidebar with a search bar and a list of entities matching filters, including CMS_Interface_9, CMS_Time, CSCL_Main, C_And_T, Conf_Table, Csci_Dbg, Message_Definition, Message_Handler, Message_Uilities, NAVS_Interface, Ressources_Definition, TA_PMP_LNAVS, T_16_Integer, T_ACC_IO_ROUGH, and T_ACC_IO_ROUGH. On the right, there are panels for 'ENTITIES DEPENDING ON ME' and 'ENTITIES I DEPEND ON', both showing a list of entities. At the bottom right, a table displays static analysis metrics:

Property	Value
Cyclomatic complexity	7
Number of children	10
Number of dead children	0
Number of external providers	4

Conclusion

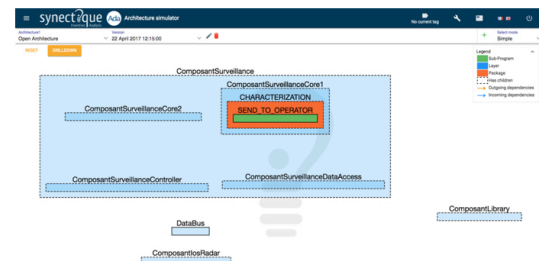
Managing the rearchitecting projet

Materializing an architecture



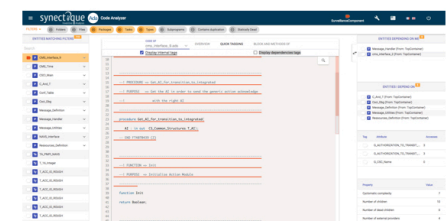
Managing the rearchitecting projet

Allocating source code



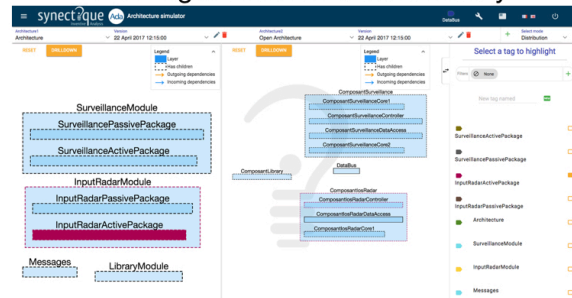
Managing the rearchitecting projet

Locating architecture in source code



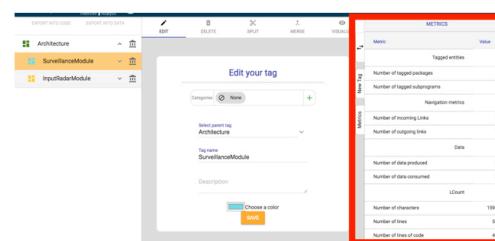
Managing the rearchitecting projet

Ensuring architecture consistency



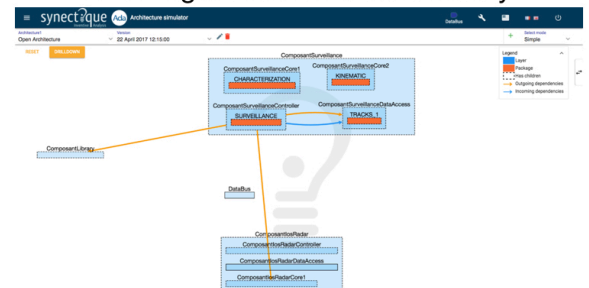
Managing the rearchitecting projet

Ensuring architecture consistency



Managing the rearchitecting projet

Ensuring architecture consistency



Future

- **Ease the tagging**
- **Applying tags on other rearchitecting projects**
- **Progress metric**