Software Architecture Compliance Checking with HUSACCT

Leo Pruijt, HU University of Applied Science, Utrecht

In cooperation with:
Raymond Slot, Wiebe Wiersema, HU, Utrecht
Christian Köppe, HAN, Arnhem
Sjaak Brinkkemper, Jan Martijn van der Werf, University Utrecht

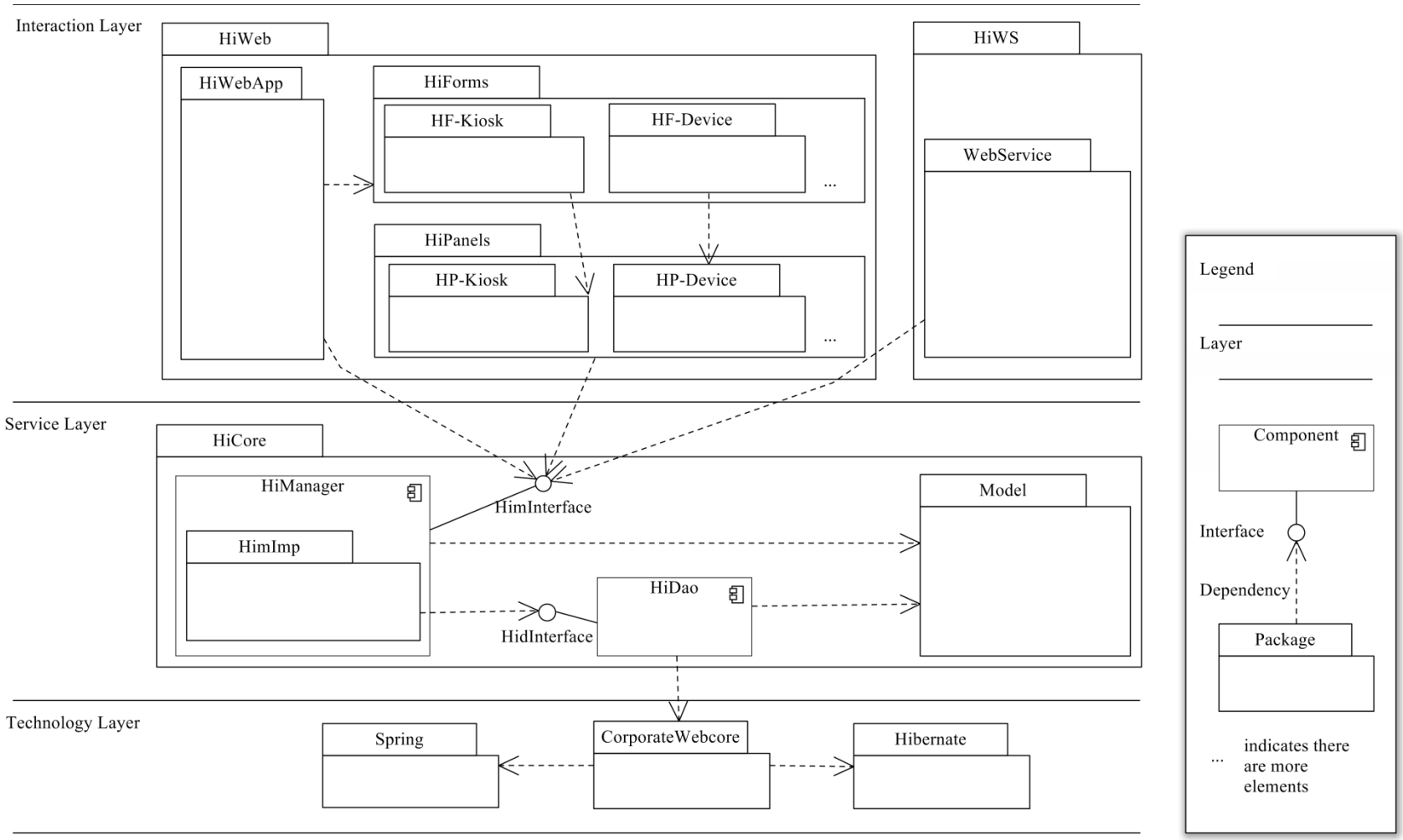
Agenda

- Software Architecture & Quality Attributes
- Architecture Compliance Checking
- HUSACCT, a tool to provide SRMA support
- Example
- Questions

Software Architecture (SA)

- Software architecture is of major importance to achieve
  - the business goals
  - functional requirements
  - quality requirements of a system
- However …
  - Architectural models tend to be of a high-level of abstraction
  - Deviations of the software architecture arise easily during
    - the development and
    - evolution of a system
- Architectural Erosion has a negative impact on the system's quality attributes

SA Example (Schiphol Group)



Common Module & Rule Types

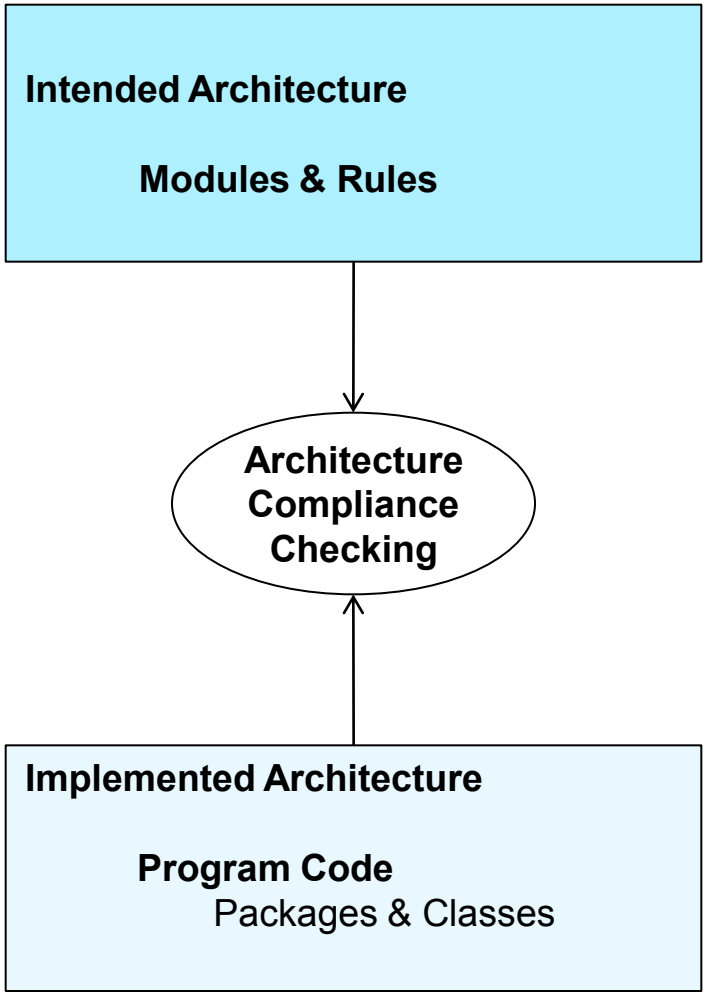
In practice …

- Modules with different semantics are used commonly
  - Subsystems, Layers, Components, Interfaces, External Systems
- Rules of different types are used
  - Some of them specific for a certain type of module
  - E.g., ‘Back call ban’ and ‘Skip call ban’ are specific for Layers

Type of Rule	Example (E)
Is not allowed to use	HiPanels is not allowed to use HiWS.
Back call ban	Service Layer is not allowed to use the Interaction Layer.
Facade convention	Component HiManager may be accessed only via HimInterface.
Is only allowed to use	HiForms is only allowed to use HiPanels.
Is the only module allowed to use	CorporateWebcore is the only module allowed to use Hibernate.

Architecture Compliance Checking (ACC)

- ACC verifies the conformance of
  - implemented program code to
  - high-level models of architectural design
- Static ACC focuses on the modular architecture
- Related quality attributes:
  - Accuracy
  - Maintainability
    - analyzability, changeability, testability
  - Portability
    - adaptability, replace ability



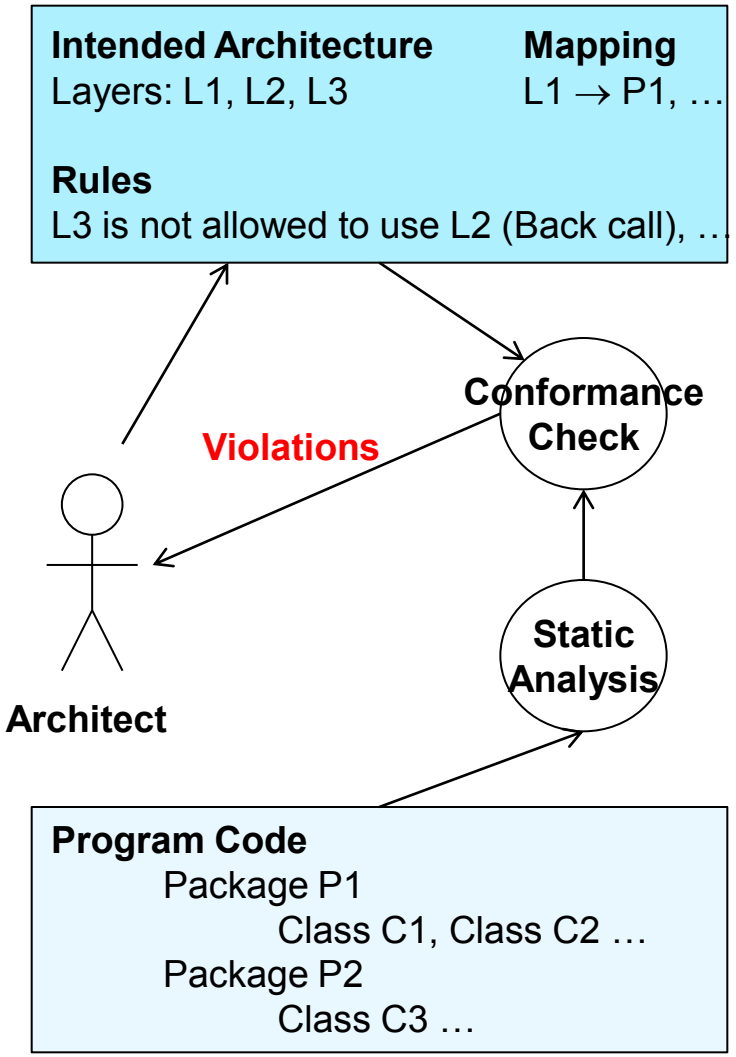
Static Architecture Compliance Checking

Currently …

- Adoption of ACC in practice is limited
- Tool support of the common sets of module & rule types is limited

Research Goals …

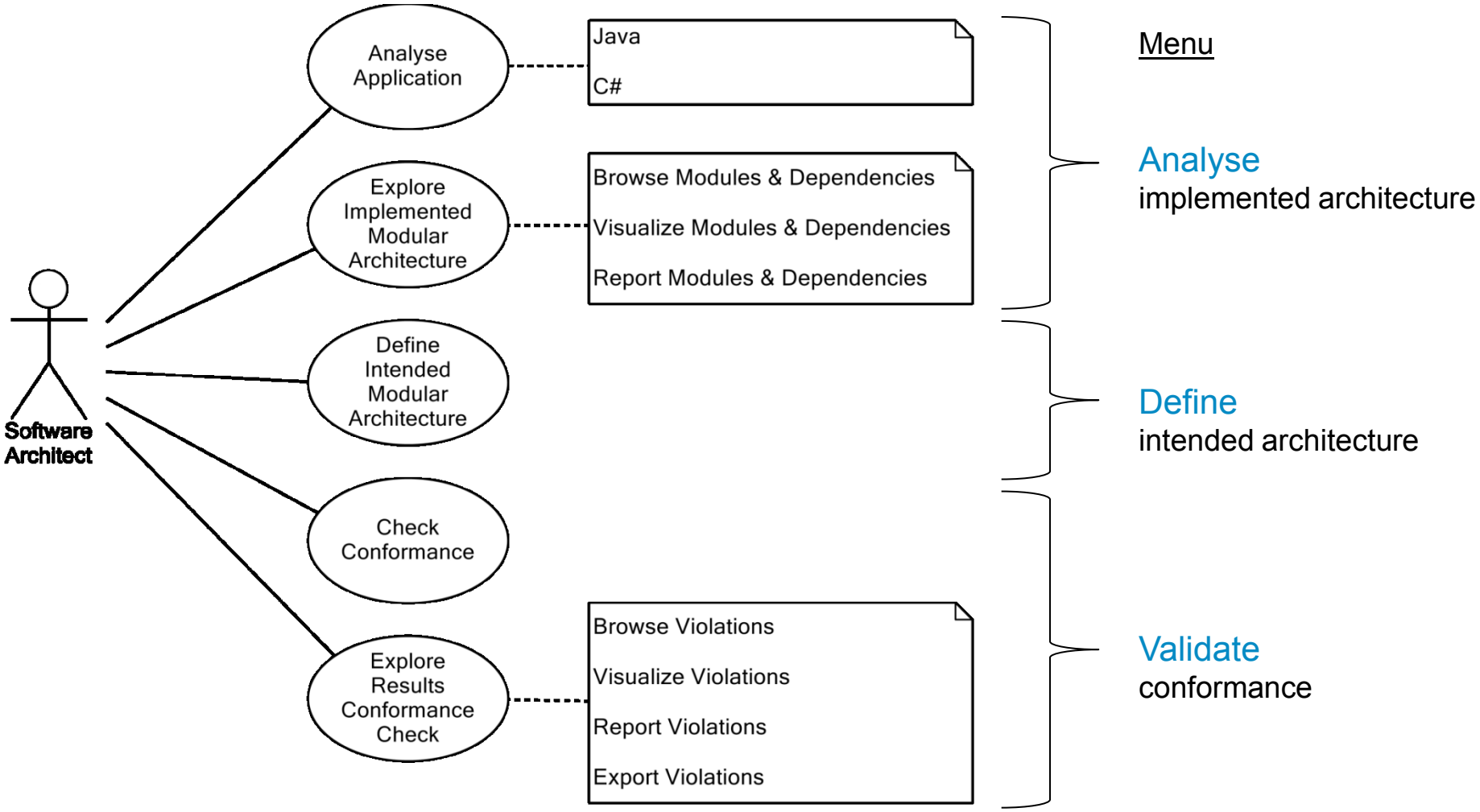
- Improve tool support
- Promote ACC in practice & education



ACC Process

1) Study the intended architecture (SAD/Architecture Notebook)
    1) Requirements
    2) Architectural decisions: Modules & Rules
    3) Mapping of modules to program code
2) Acquire additional or missing information
3) Enter intended architecture in ACC-tool
    1) Modules & Rules
    2) Mapping of modules to program code
4) Run conformance check
5) Study and discuss results
    1) Violations
    2) Relevance

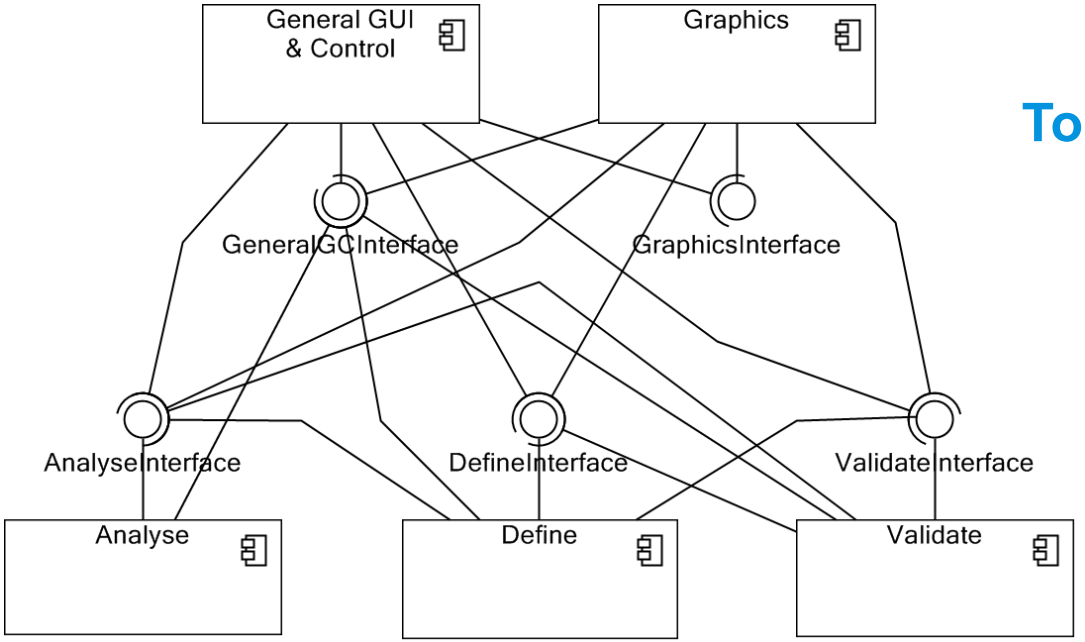
HUSACCT: Hogeschool Utrecht
Software Architecture Compliance Checking Tool



Outstanding Characteristics

- HUSACCT is free-to-use & open source
  - Download, video & instruction at http://husacct.github.io/HUSACCT/
- Support of rich sets of Module and Rule Types
  - 5 common Module Types with different semantics
    - Subsystem, Layer, Component, Interface, External system
  - 11 common Rule Types
- Extensive Semantic Support of the Module and Rule Types, e.g.:
  - Automatic creation of default rules, according to the Module Type
  - Type of Module determines which Rule Types are selectable
- Configurable support
  - Enable/Disable rules, Exception rules, Default rule configuration

ACC Example:
Modular Architecture HUSACCT_1.0

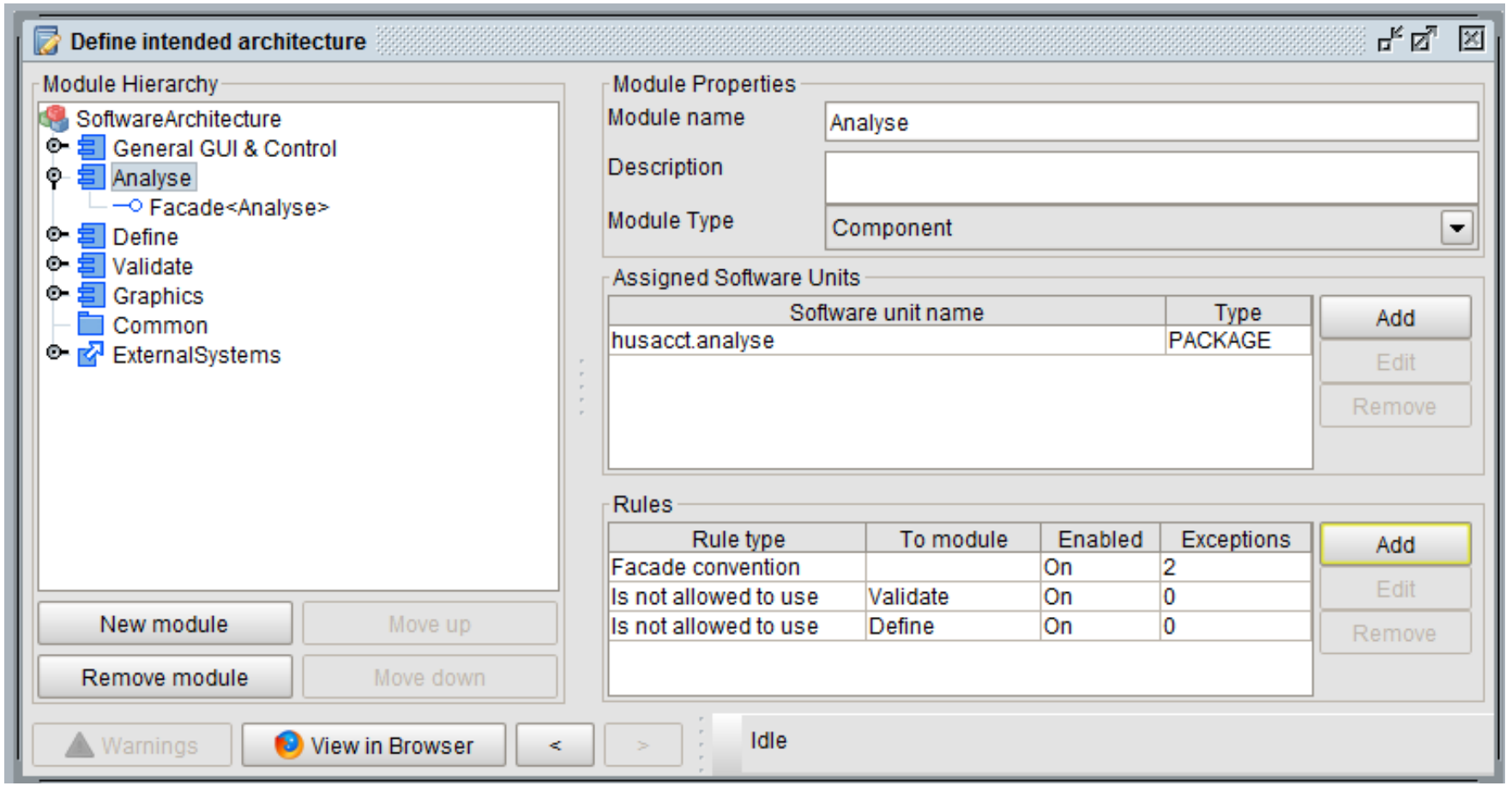


Top-level Components

Rules

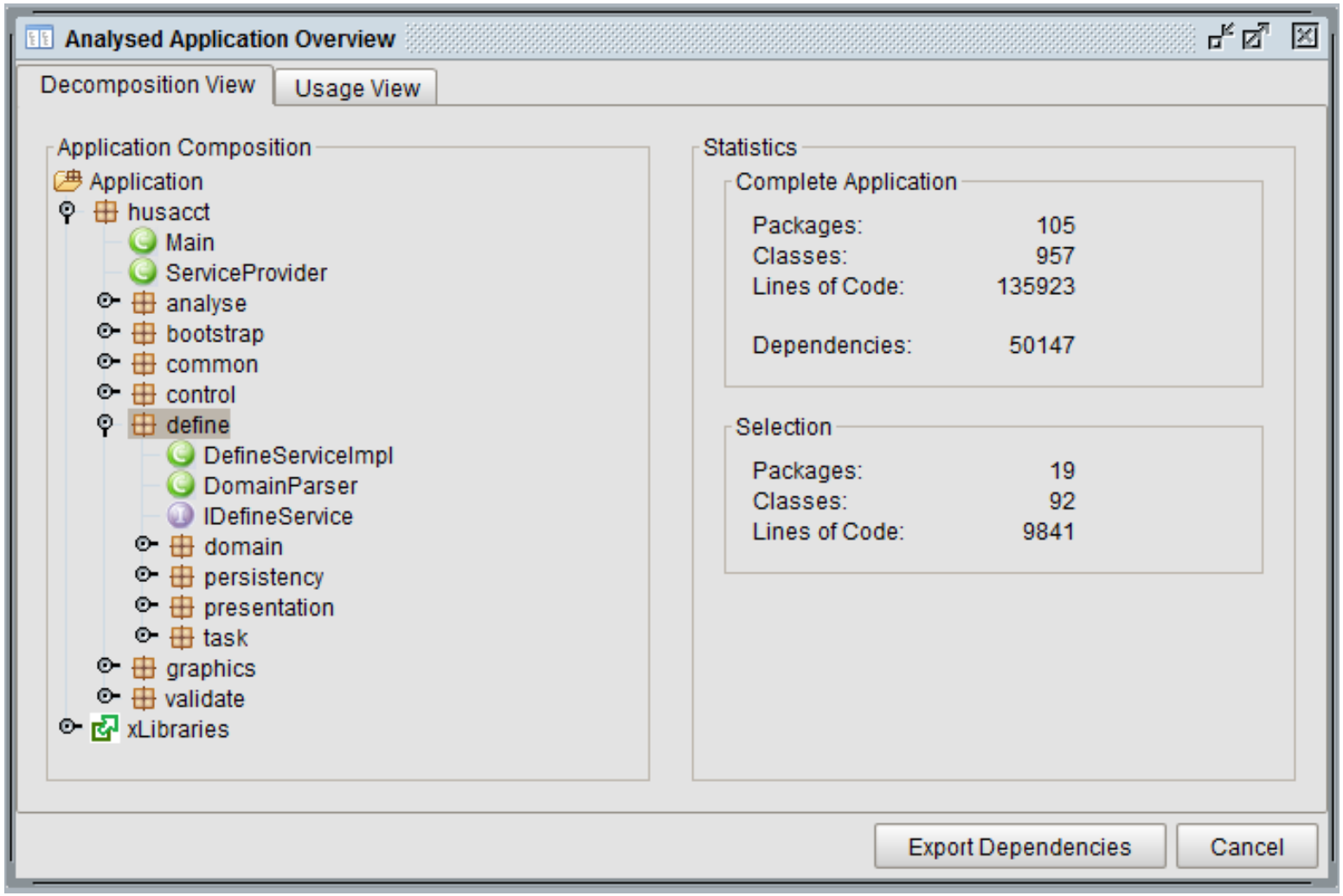
From-Module	Constraint	To-Module
Analyse	is not allowed to use	Define
Analyse	is not allowed to use	Validate
General GUI & Control	Is the only module allowed to use	Graphics
All five components	Facade convention	

Define Intended Architecture



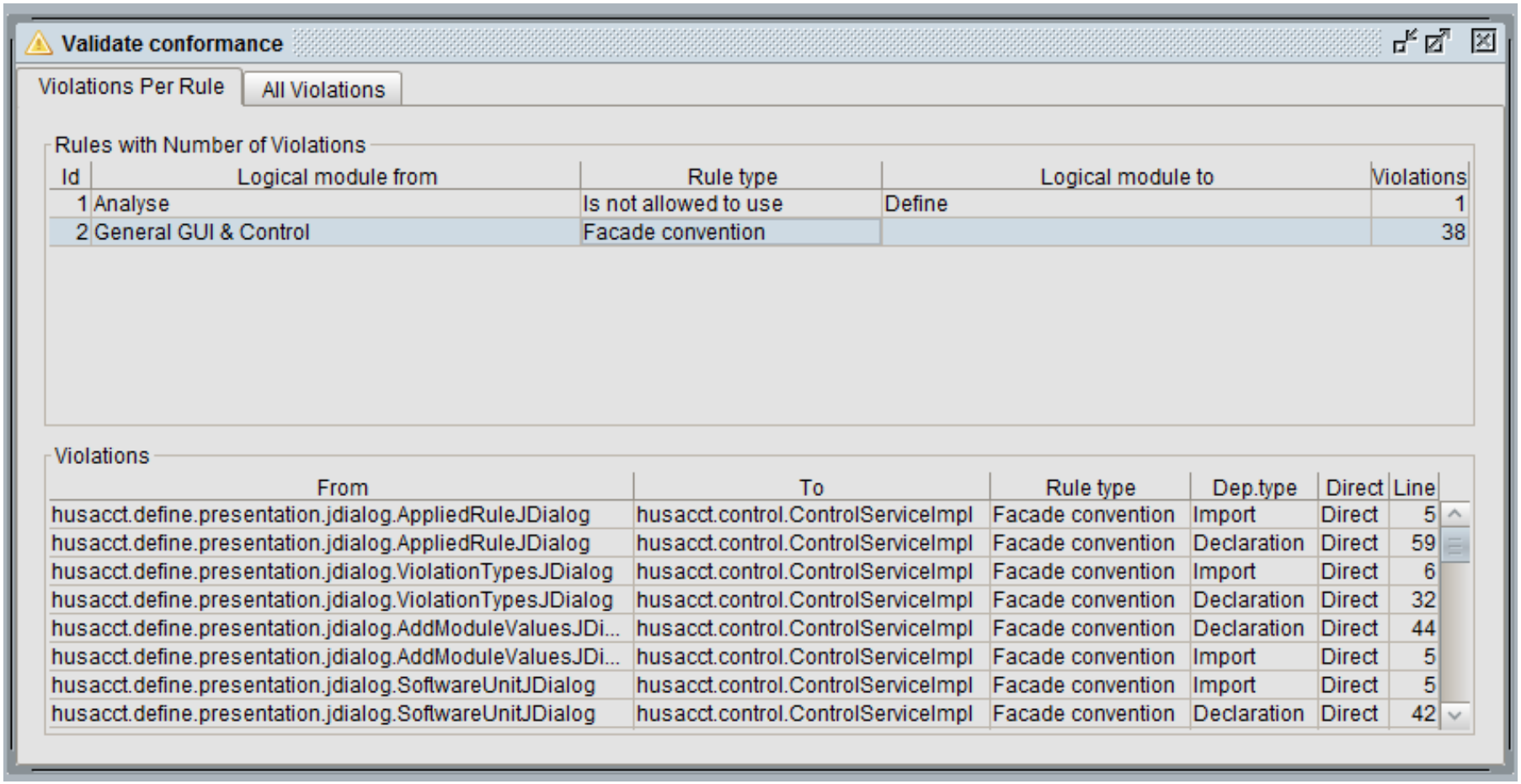
Analyse Implemented Architecture

Needed to Assign Software Units and to Check Conformance



Validate Conformance

Results of the Conformance Check



Validate conformance

Violations Per Rule | All Violations

Rules with Number of Violations

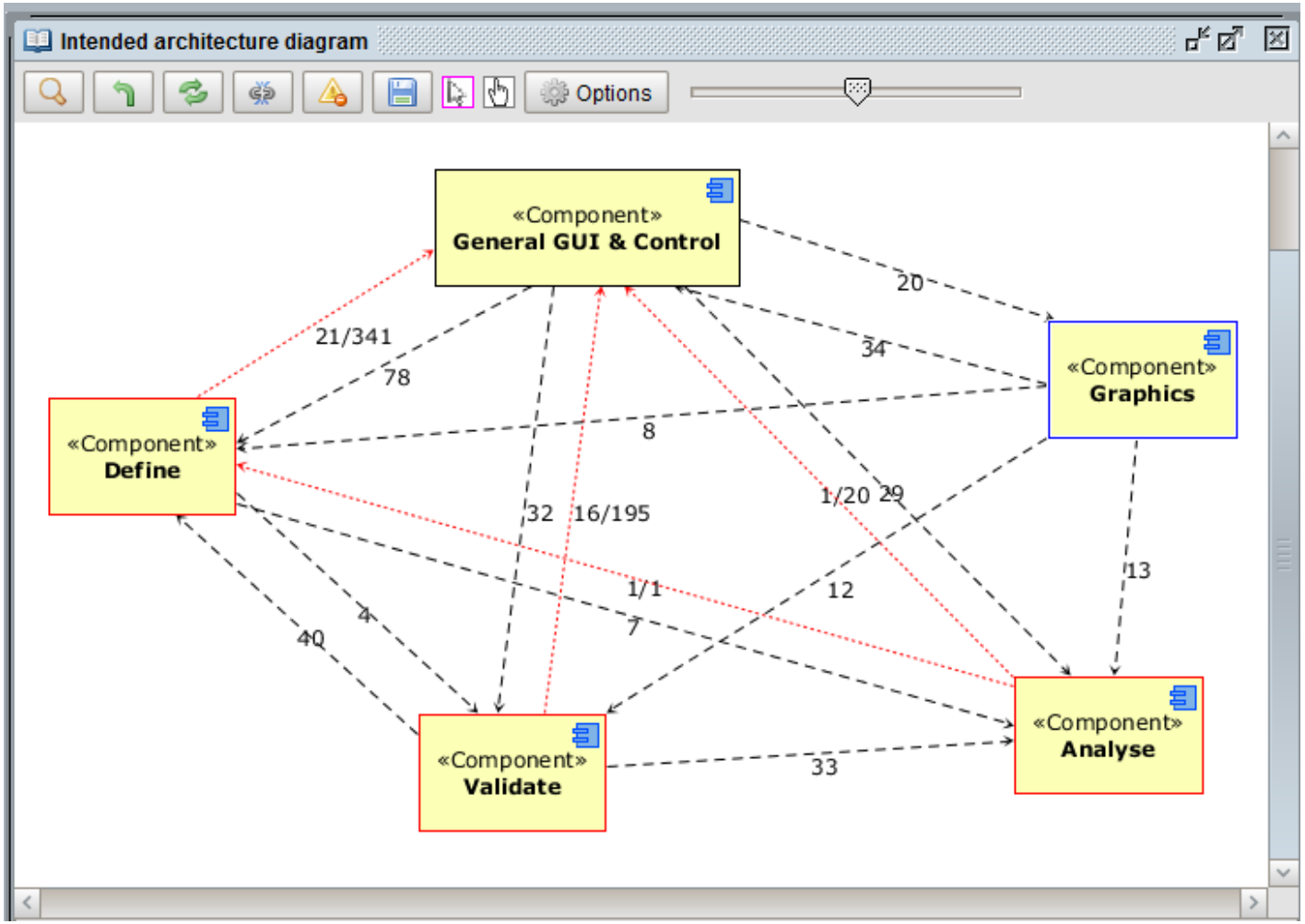
Id	Logical module from	Rule type	Logical module to	Violations
1	Analyse	Is not allowed to use	Define	1
2	General GUI & Control	Facade convention		38

Violations

From	To	Rule type	Dep.type	Direct	Line
husacct.define.presentation.jdialog.AppliedRuleJDialog	husacct.control.ControlServiceImpl	Facade convention	Import	Direct	5
husacct.define.presentation.jdialog.AppliedRuleJDialog	husacct.control.ControlServiceImpl	Facade convention	Declaration	Direct	59
husacct.define.presentation.jdialog.ViolationTypesJDialog	husacct.control.ControlServiceImpl	Facade convention	Import	Direct	6
husacct.define.presentation.jdialog.ViolationTypesJDialog	husacct.control.ControlServiceImpl	Facade convention	Declaration	Direct	32
husacct.define.presentation.jdialog.AddModuleValuesJDi...	husacct.control.ControlServiceImpl	Facade convention	Declaration	Direct	44
husacct.define.presentation.jdialog.AddModuleValuesJDi...	husacct.control.ControlServiceImpl	Facade convention	Import	Direct	5
husacct.define.presentation.jdialog.SoftwareUnitJDialog	husacct.control.ControlServiceImpl	Facade convention	Import	Direct	5
husacct.define.presentation.jdialog.SoftwareUnitJDialog	husacct.control.ControlServiceImpl	Facade convention	Declaration	Direct	42

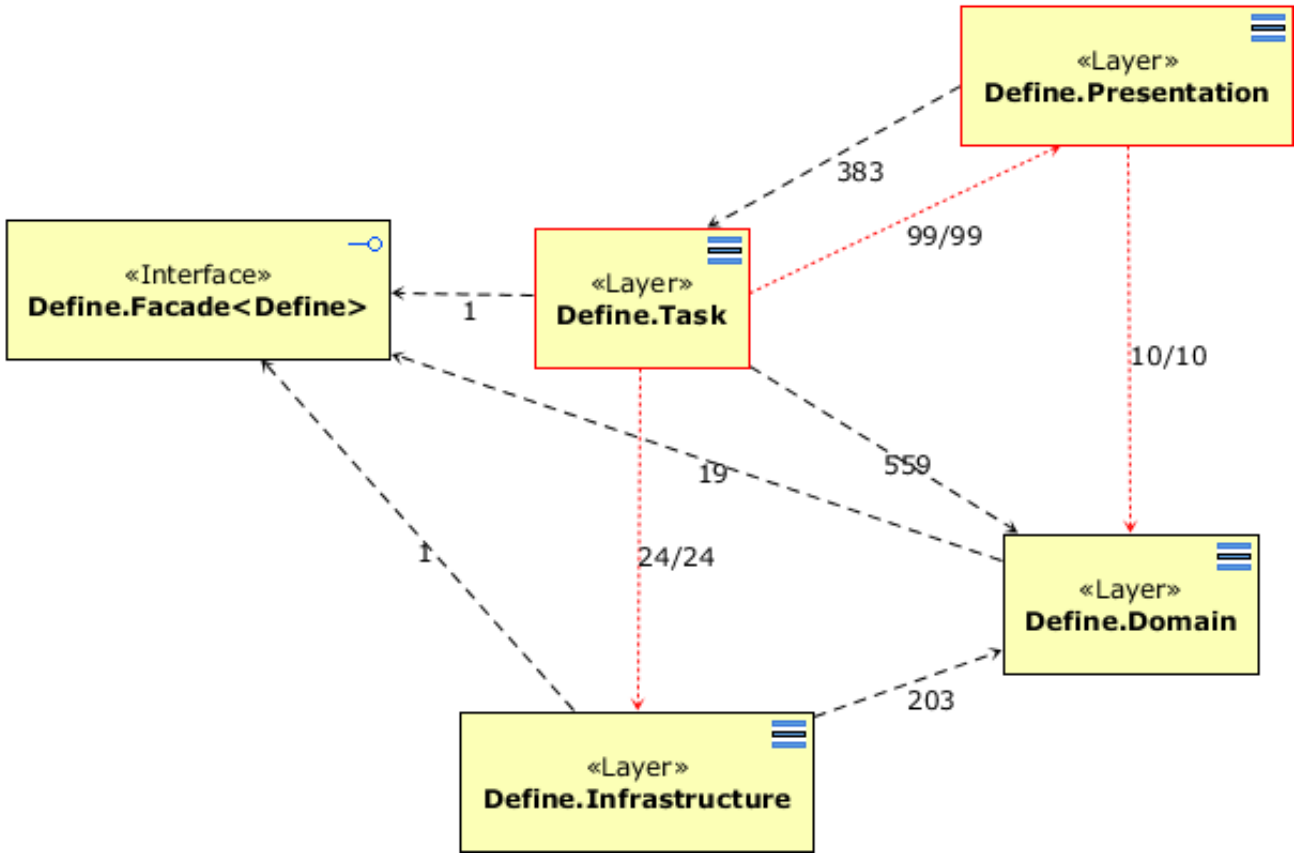
Intended Architecture Diagram

With the intended top-level components, their dependencies and violations



Intended Architecture Diagram

With the intended layers within Define, their dependencies and violations



Status and Outlook

- **HUSACCT_3.4** is fit for practical use
  - Accurate, fast, easy in use
- Intended use:
  - Introduction of ACC within organizations
  - Software architecture education
    - Relate abstract models to code
  - Stimulation of tool vendors to provide support for Semantically Rich Modular Architectures
- Future work:
  - Case studies: Are you interested in a free ACC?
  - Extension of functionality
  - Metrics

Finally

- More information:
  - Ask me: Leo Pruijt, leo.pruijt@hu.nl
  - Watch the video at http://husacct.github.io/HUSACCT/
  - Read the published papers
- Thank you for your attention!
- Questions?

1) HUSACCT: Architecture Compliance Checking with Rich Sets of Module and Rule Types. *2014 IEEE/ACM Int. Conf. on Automated Software Engineering, ASE 2014-09, Vasteras, Sweden*

2) Architecture Compliance Checking of Semantically Rich Modular Architectures: A Comparative Study of Tool Support. *Int. Conf. on Software Maintenance, ICSM 2013-09, Eindhoven, NL*

3) A Metamodel for the Support of Semantically Rich Modular Architectures in the Context of Architecture Compliance Checking. *SAEroCon workshop, WICSA 2014-04, Sydney, Austratlia*

4) On the Accuracy of Architecture Compliance Checking Support: Accuracy of Dependency Analysis and Violation Reporting. *Int. Conf. on Program Comprehension, ICPC 2013-05, San Francisco, USA*