

How-to Use Software Architecture Models

- Purpose
- Use case
- Import an Architecture Checker model
- Test a model against an application
- View and edit models
- Moving a model to production

Purpose

The purpose of this How-To doc is to explain how to import Architecture Checker models into AIP Console and to use them to detect application structural flaw.

The different actions explained here have been done with **AIP Console 1.9.0**.

Use case

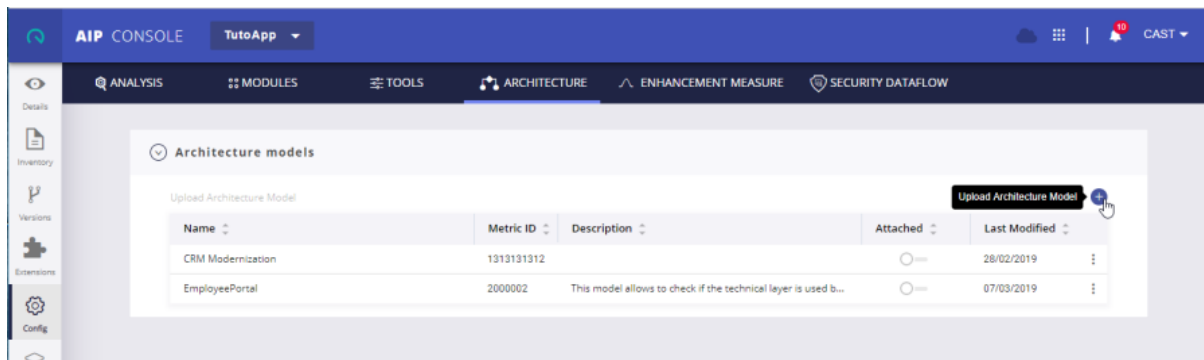
As an **Application Owner** you on-boarded an application in AIP Console and you want to check if the software architecture that has been designed by **Software Architects** is followed. This can be done by using Architecture Checker and, for that, you must have analyzed a version (and generated a snapshot) and you need an architecture model.


Software Architects may have one that corresponds to the software architecture to be verified. If not, then a new one can be created with the client-server version of Architecture Checker as it is not yet possible to do that in AIP Console.

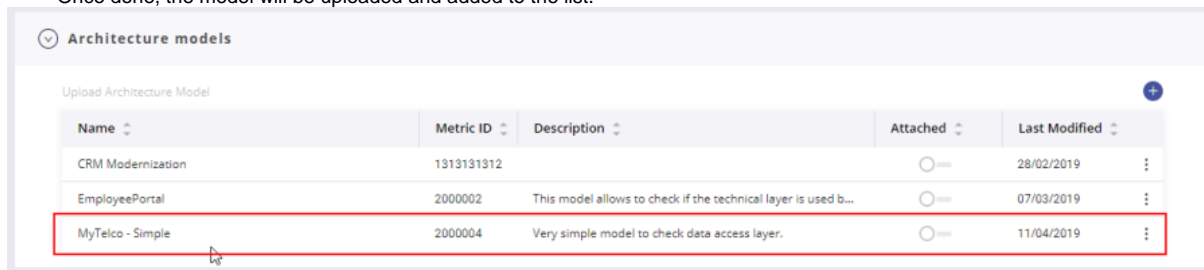
Import an Architecture Checker model

To import an Architecture Checker model in AIP Console.

1. Go to the application **Config** page
2. Click the **ARCHITECTURE** tab. The displayed page shows the models that might have been previously uploaded.



3. To add a new model to that list, click the  button. A file selection dialog opens and lets you to select a model. Once done, the model will be uploaded and added to the list.



However, before importing a model, ensure this one has been assigned to a valid ID. Otherwise, this will prevent importing that model. You can refer to the product documentation for more details: [AIP Console - Architecture](#).

i Note that in v. 1.20, you may find **.castrule** files are available for selection. These files should not be added as they do not contain Architecture Models (they contain details of Quality Rule models).

Test a model against an application

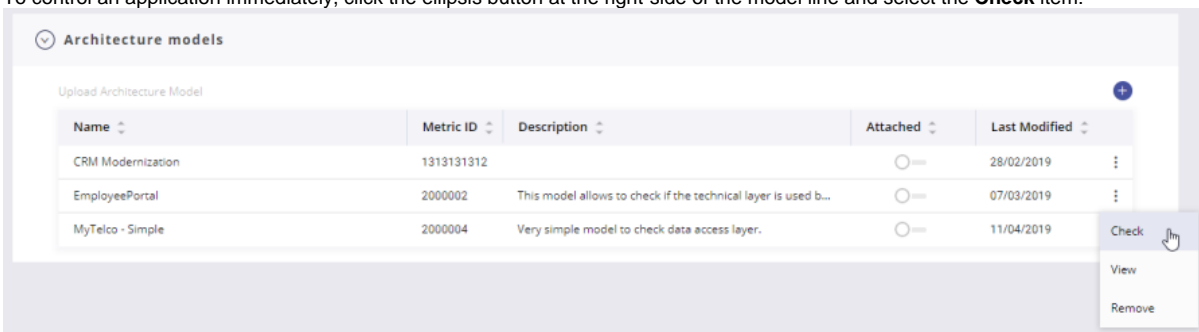
Before attaching a model to an application, you have to validate if it is the right one to find structural flaws.

An Architecture Checker model is mainly based on layers and dependencies. Layers capture objects that should belong to a given software layer. Dependencies specify which links are authorized between objects belonging to different layers and which ones are not.

For instance, a model can define 3 layers: Database, DB Access, and Front, with authorized layers dependencies between Front and DB Access and between DB Access and Database. It means that objects belonging to Front layer cannot be linked to objects belonging to the Database layer. Only objects belonging to the DB Access layers can have such links.

i An Architecture Checker model is typically designed by an Software Architect.

To control an application immediately, click the ellipsis button at the right-side of the model line and select the **Check** item:



AIP Console opens a dialog and applies the Architecture Checker model against the application. As a result, the dialog shows the list of links that violate the model:

(167) Violations for architecture model: EmployeePortal

Caller Layer	Caller Name	Caller Type	Caller Fullname	Callee Layer	Callee Name	Callee Type	Callee Fullname	Link Type
EP API layer	APIACC	Cobol Program	[C:\CASTMS\Deploy...	Unassigned	ACCOUNT	Cobol CopyBook	[Unknown\Copybook...	Include
EP API layer	APIACC	Cobol Program	[C:\CASTMS\Deploy...	Unassigned	CPACCPRO	Cobol CopyBook	[C:\CASTMS\Deploy...	Include
EP API layer	APIACC	Cobol Program	[C:\CASTMS\Deploy...	Unassigned	SQLCA	Cobol CopyBook	[C:\CASTMS\Deploy...	Include
EP API layer	APIACTVT	Cobol Program	[C:\CASTMS\Deploy...	Unassigned	ACTIVITY	Cobol CopyBook	[C:\CASTMS\Deploy...	Include
EP API layer	APIACTVT	Cobol Program	[C:\CASTMS\Deploy...	Unassigned	SQLCA	Cobol CopyBook	[C:\CASTMS\Deploy...	Include
EP API layer	APIASV22	Cobol Program	[C:\CASTMS\Deploy...	Unassigned	EMP	Cobol CopyBook	[C:\CASTMS\Deploy...	Include
EP API layer	APIASV22	Cobol Program	[C:\CASTMS\Deploy...	Unassigned	SQLCA	Cobol CopyBook	[C:\CASTMS\Deploy...	Include
EP API layer	APIASV29	Cobol Program	[C:\CASTMS\Deploy...	Unassigned	EMP	Cobol CopyBook	[C:\CASTMS\Deploy...	Include
EP API layer	APIASV29	Cobol Program	[C:\CASTMS\Deploy...	Unassigned	SQLCA	Cobol CopyBook	[C:\CASTMS\Deploy...	Include
EP API layer	APICUSTO	Cobol Program	[C:\CASTMS\Deploy...	Unassigned	CUSTOMER	Cobol CopyBook	[Unknown\Copybook...	Include

10 / page

OK

View and edit models

If you want to view how the model is designed to detect structural flaws, then you can open the model editor. Edit model page allows you to make changes.

To open the model editor:

1. Click the ellipsis button at the right-side of the model line and select the **View** item.

Architecture models

Upload Architecture Model

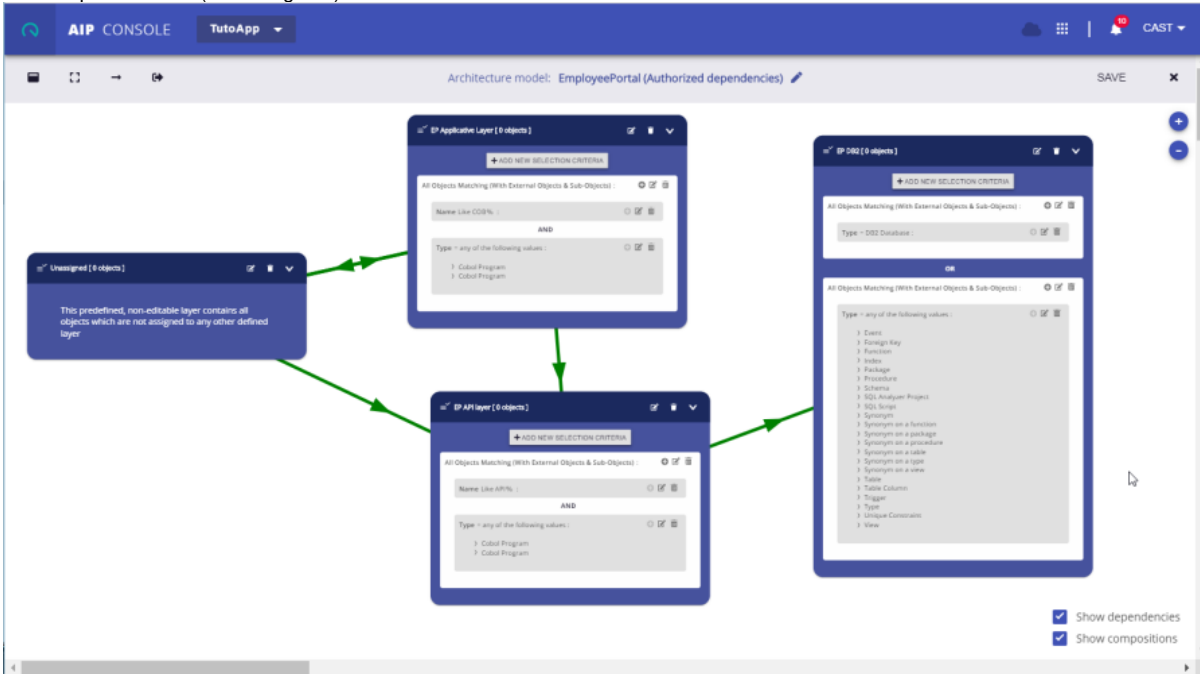
Name	Metric ID	Description	Attached	Last Modified	
CRM Modernization	1313131312		<input type="radio"/>	28/02/2019	⋮
EmployeePortal	2000002	This model allows to check if the technical layer is used b...	<input type="radio"/>	07/03/2019	⋮
MyTelco - Simple	2000004	Very simple model to check data access layer.	<input type="radio"/>	12/04/2019	⋮

Check

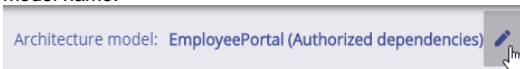
View

Remove

2. The model editor opens and displays the layers (in blue) and sets (in green) that define the model, with the associated dependencies (in bold green) and composition links (in dotted green).



3. If you want to see or maybe update the model properties (name, ID, documentation, weight, and so on), then simply click the edit button near the model name:



4. The dialog box displays these properties:

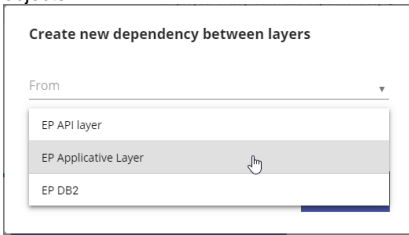
Edit Architecture Model: EmployeePortal

MODEL	DOCUMENTATION	ASSESSMENT MODEL
<p>Rationale Using the technical layer makes application programs independent from the physical data storage.</p> <p>Remediation Please use the API programs to access databases.</p> <p>Reference Software Architecture Team</p> <p>Sample</p> <p>Remediation Sample</p> <p>Scope COBOL application programs</p> <p>Output List of unexpected DB accesses done by COBOL application programs</p>		
<p>CANCEL</p>		<p>UPDATE</p>

5. You can add new **Layers**, **Sets**, and **Dependencies** by using the tool bar.

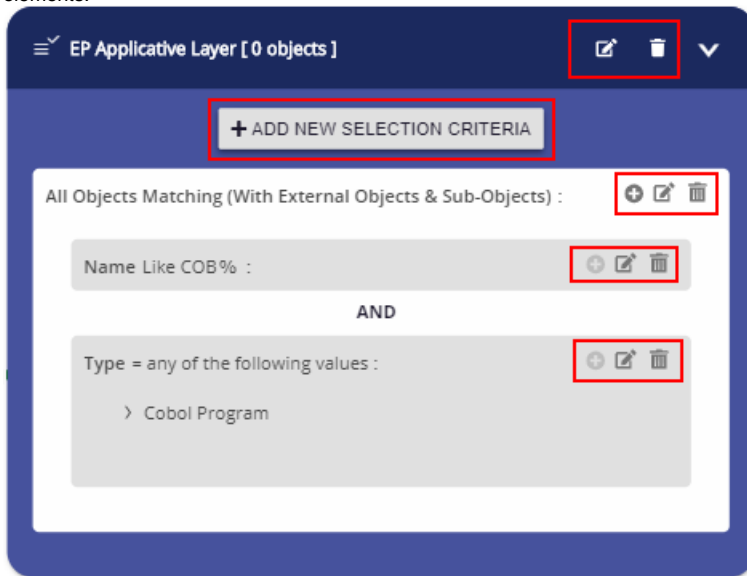


6. From left to right, you can add a new Layer object to the model, you can add a new Set object, you can add a new Dependency between 2 Layer objects



You can remove an existing Dependency from the model.

To change the selection criteria defined to capture objects in layers and sets, use the **Add, Edit, Remove** icons associated to the different graphical elements:



Save your changes before exiting the model editor. Even Layer and Set objects position will be saved.

Moving a model to production

Architecture Checker models are used to control the application implementation. For that, a model must be attached to the considered application and a snapshot must be produced. Attaching a model means a new dedicated rule is added to the Assessment Model with the contribution specified in the model properties.

The rule is violated when there is at least a link that does not respect the dependencies defined in the model.

The below screen shot shows the rule generated from the above Architecture Checker model with the violations that have been detected:

The screenshot displays the 'Engineering Dashboard - HR Management' interface. At the top, it shows a snapshot from 2019-04-15, version 3, and date 2019-04-15. The breadcrumb trail indicates the current view is 'Architecture Check: EmployeePortal'. The 'Rules...' section shows a summary for 'Architecture Check: EmployeePortal' with 0 critical, 0 high, 109 medium, and 0.00% low severity items, and 6 violations. The 'Violations' table lists 8 items, all with a risk of 'n/a' and a status of 'Unchanged'.

OBJECT NAME	LOCATION	RISK	STATUS
[C:\CASTMS\Deploy_Webi_8310\HR ... Management\main_sources\Mainframe\API]APIACC		n/a	Unchanged
[C:\CASTMS\Deploy_Webi_8310\HR ... nagement\main_sources\Mainframe\API]APIACTVT		n/a	Unchanged
[C:\CASTMS\Deploy_Webi_8310\HR ... nagement\main_sources\Mainframe\API]APIASV22		n/a	Unchanged
[C:\CASTMS\Deploy_Webi_8310\HR ... nagement\main_sources\Mainframe\API]APIASV29		n/a	Unchanged
[C:\CASTMS\Deploy_Webi_8310\HR ... nagement\main_sources\Mainframe\API]APICUSTO		n/a	Unchanged
[C:\CASTMS\Deploy_Webi_8310\HR ... nagement\main_sources\Mainframe\API]APITP014		n/a	Unchanged
[C:\CASTMS\Deploy_Webi_8310\HR ... nagement\main_sources\Mainframe\API]APITP015		n/a	Unchanged
[C:\CASTMS\Deploy_Webi_8310\HR ... nagement\main_sources\Mainframe\API]APITP016		n/a	Unchanged
[C:\CASTMS\Deploy_Webi_8310\HR ... nagement\main_sources\Mainframe\API]APITP017		n/a	Unchanged
[C:\CASTMS\Deploy_Webi_8310\HR ... nagement\main_sources\Mainframe\API]APITP018		n/a	Unchanged

You should manage models carefully. If you move a model to production (i.e. you attached it to an application), then we recommend to not change it, detach it, or remove it without having evaluated the impact on dashboard results.

For instance, if you change the ID, then the rule with the previous ID will disappear and a new one with the new ID will appear. If you change the weight or the critical attribute, then the grade will be impacted. Changing the selection criteria may impact the layer content and therefore the rule results.

Move only the stable models to production.

Changing the model documentation does not impact the rule results but can increase the understanding regarding the model checks.