



# Verification Plan in Requirements Management Tool: Simple Traceability and Automated Interface to Regression Manager

Jan Kreisinger, Sanjay Chatterjee

Allegro MicroSystems



# Verification Plan in Requirements Management Tool:

Simple Traceability and Automated Interface to Regression Manager

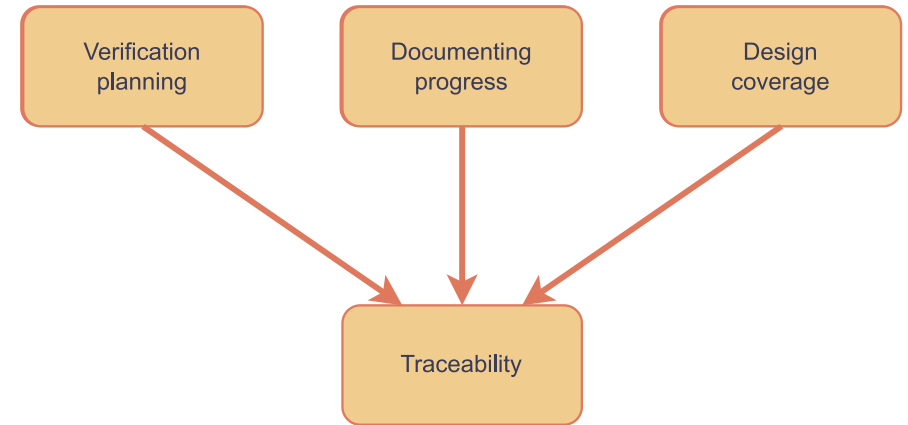
- How to avoid missing a change in the requirements?
- How to make sure my verification plan is complete?
- How to measure and present verification progress?
- How to automate data transfer between different tools?



# Agenda

- Traceability concept
- Verification workflow efficiency
- Data organization in requirements management (RM) tool
- Interface automation
- Traceability analysis
- Beyond Jama Connect and Verisium Manager
- Summary

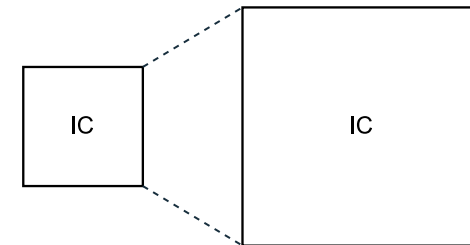
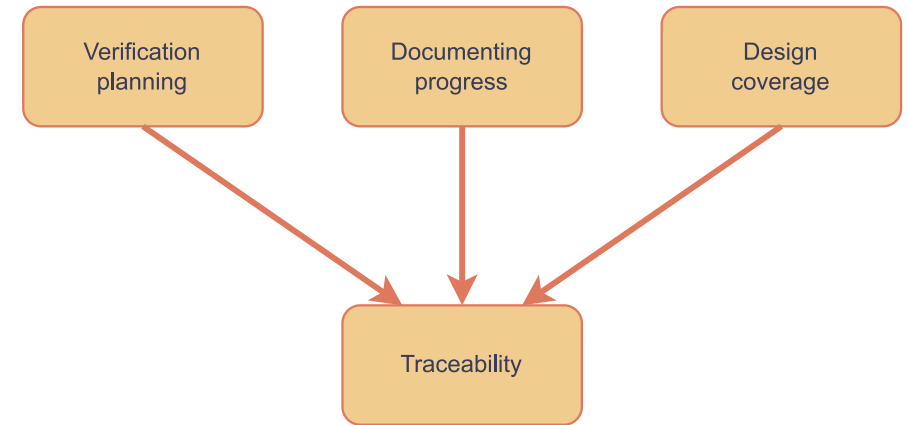
# Why traceability?



# Why traceability?

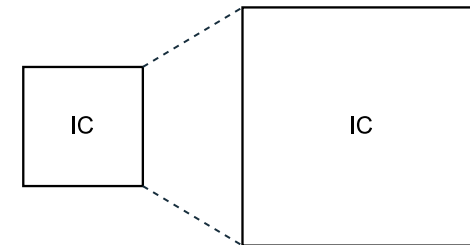
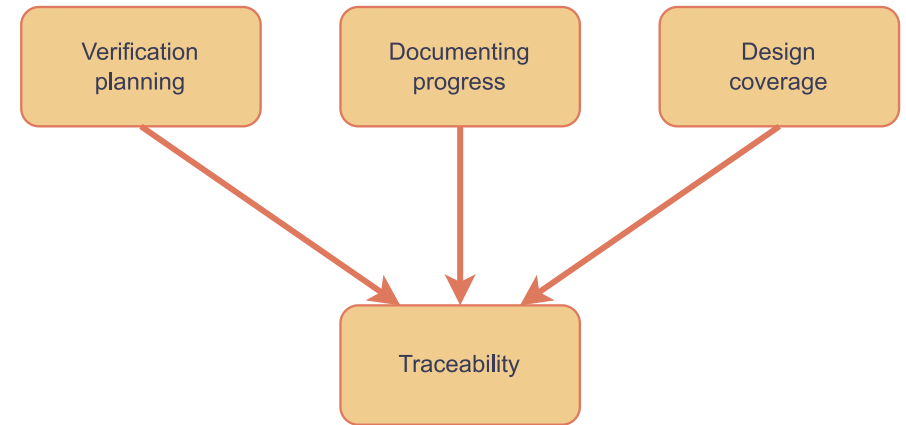
- Challenges:

- Increasing complexity
- Functional safety
- Design requirements change
- Traceability across multiple tools



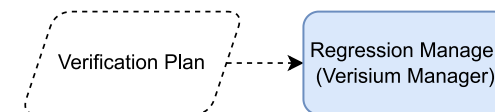
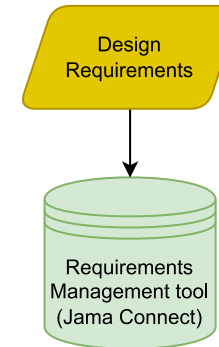
# Why traceability?

- Challenges:
  - Increasing complexity
  - Functional safety
  - Design requirements change
  - Traceability across multiple tools
- Incomplete traceability leads to:
  - Project schedule slip
  - Incomplete verification results
  - Missed bugs

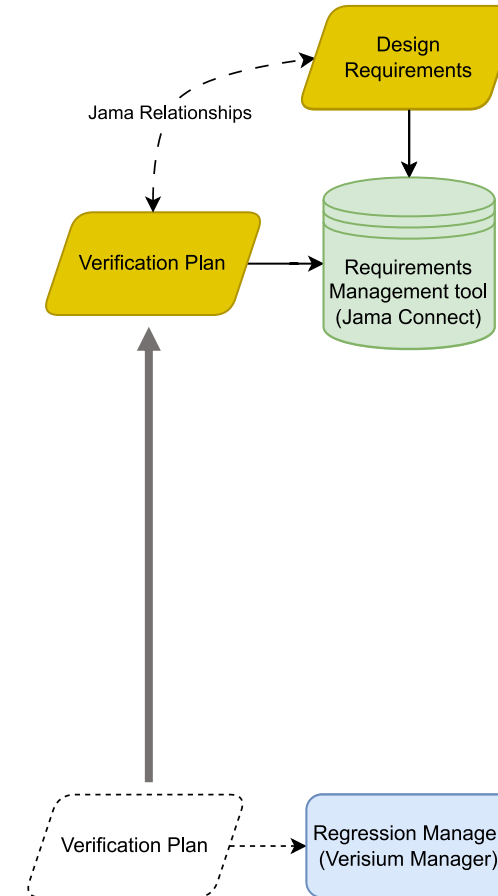


# Verification workflow efficiency

- Includes requirements tracing
- Project status overview
- User friendly
- License usage optimization
- Automzied interface and repetitive tasks



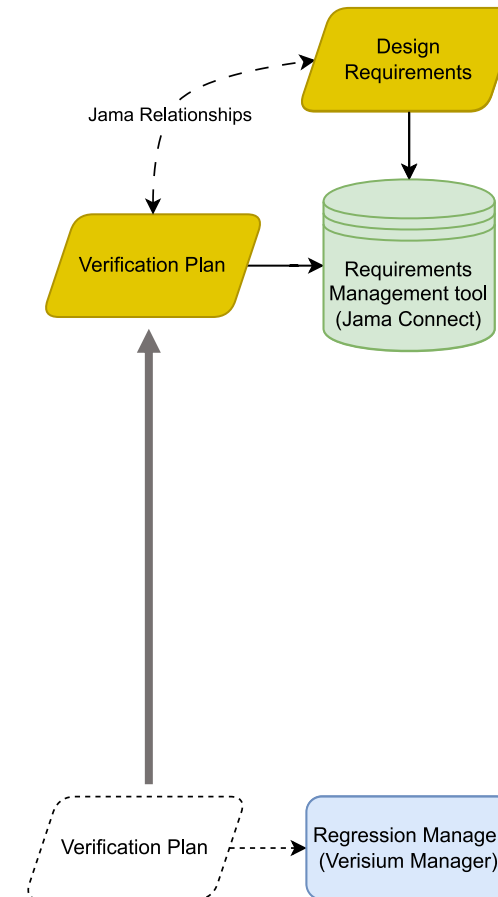
# Data organization in the RM tool



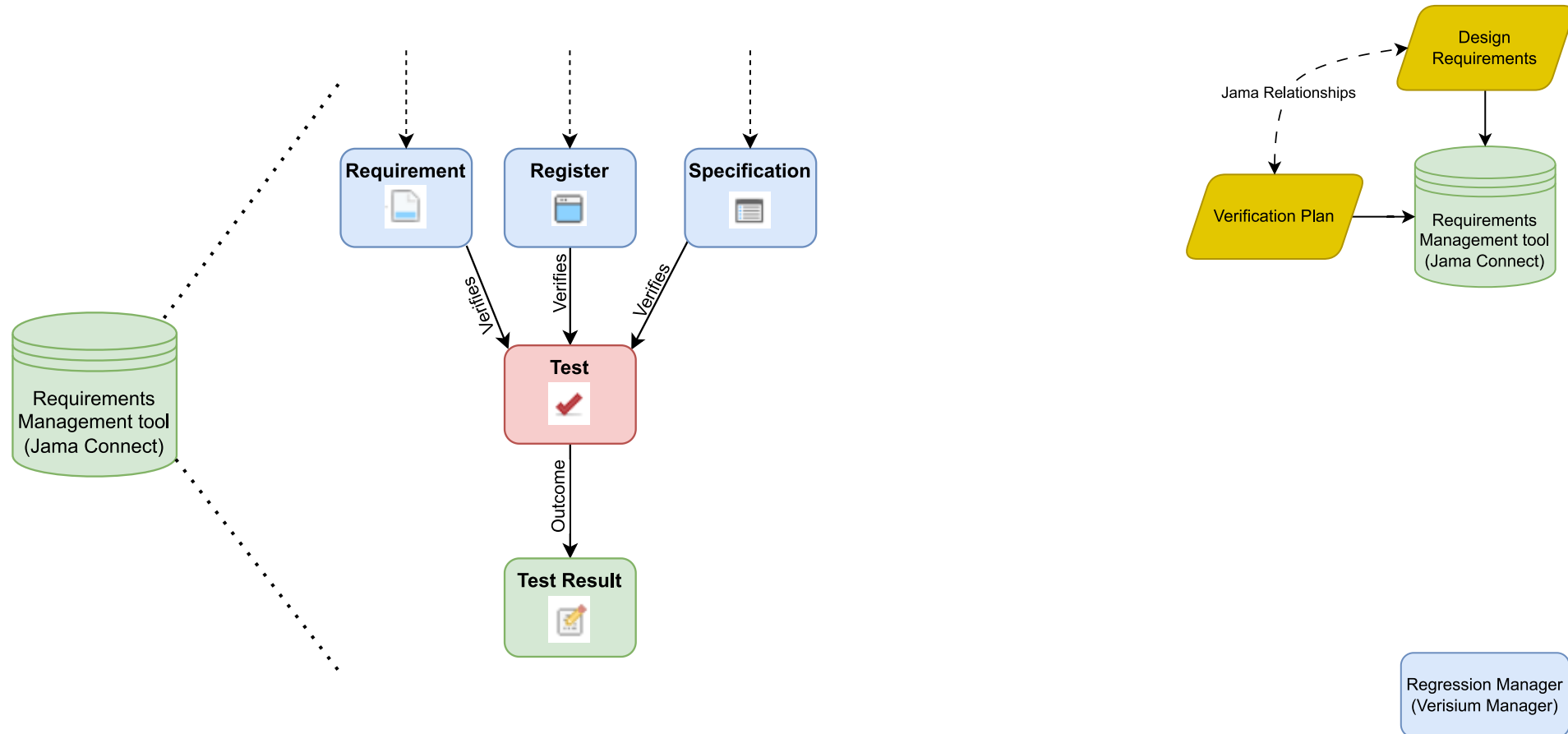


# Data organization in the RM tool

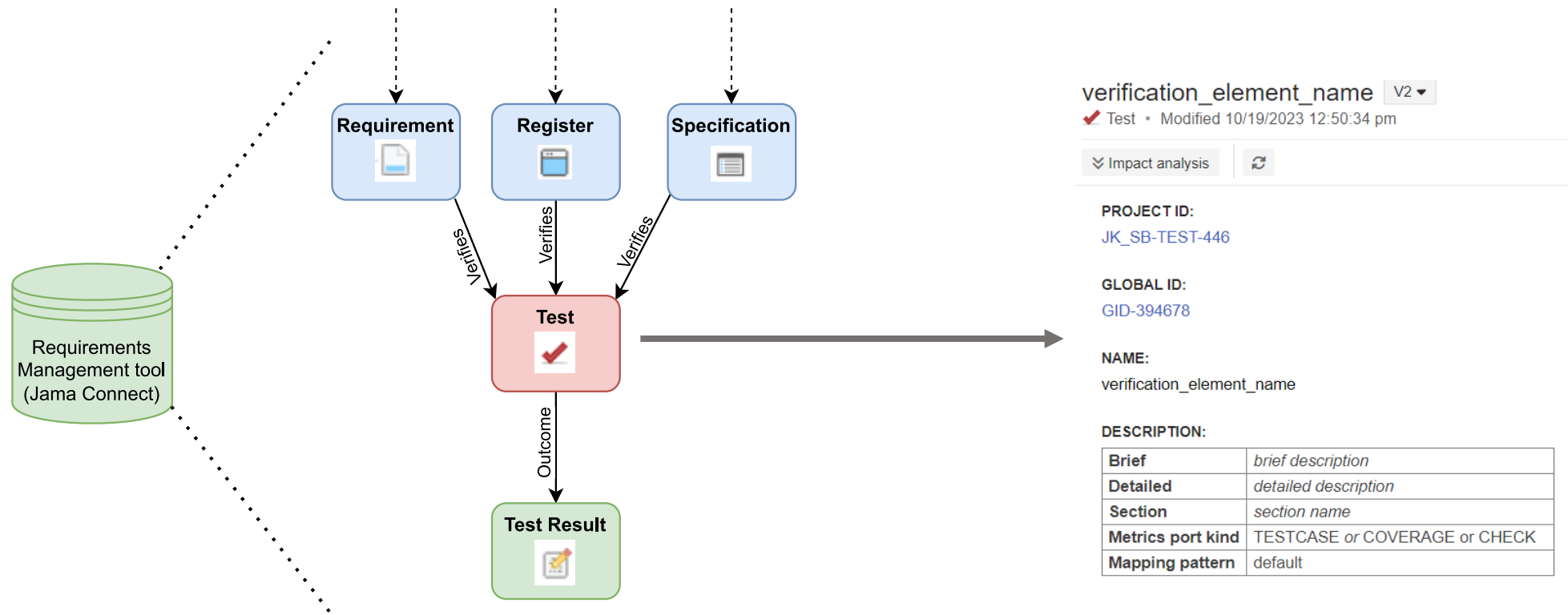
- Verification plan in RM tool
  - Direct requirements mapping
  - Accessible to whole development team
  - Easy change impact analysis
  - Reduced Verisium Manager license usage



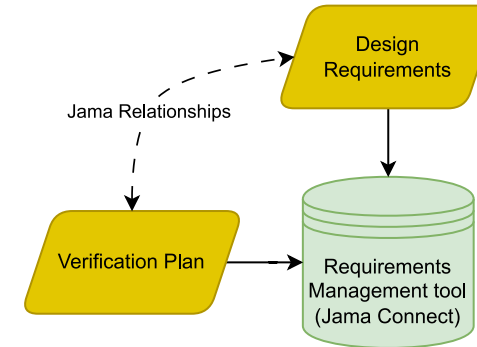
# Data organization in the RM tool



# Data organization in the RM tool



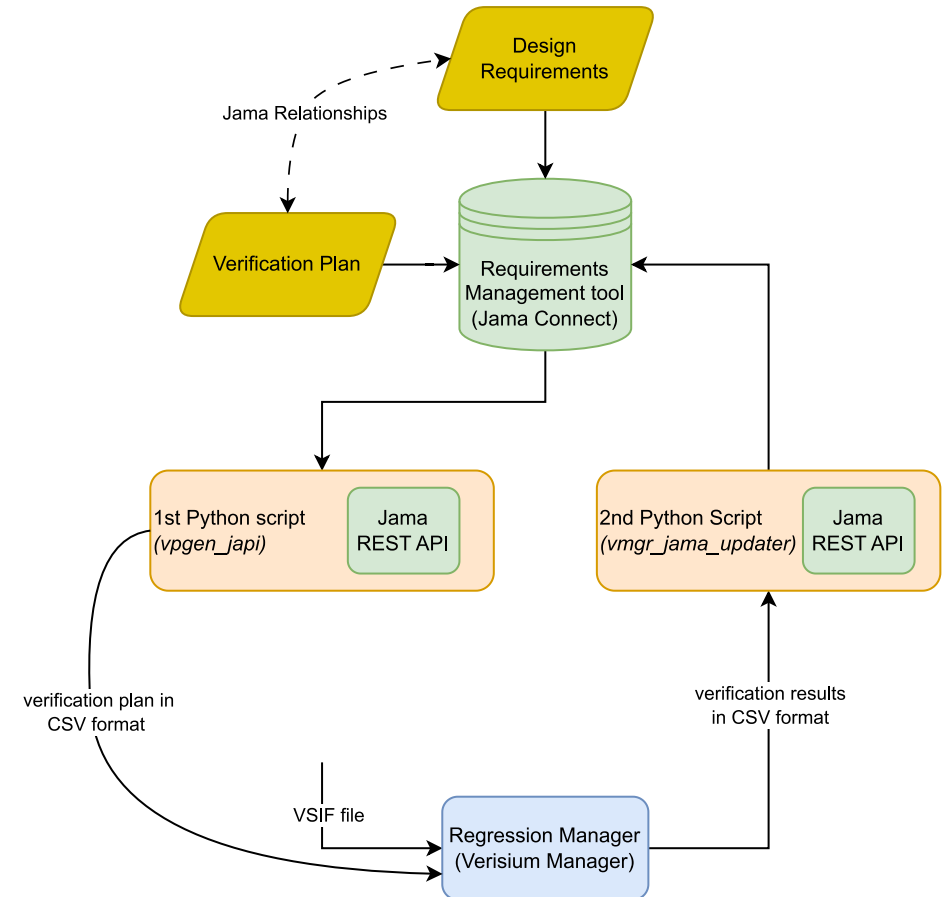
# Interface automation



Regression Manager  
(Verisium Manager)



# Interface automation



# Interface automation

## Regression session configuration:

Components / DVCON / Verification Plan / JK\_SB-TXT-289

vsif\_cfg V5

Text • Modified 10/23/2023 01:09:16 pm

Impact analysis

API ID:  
633636

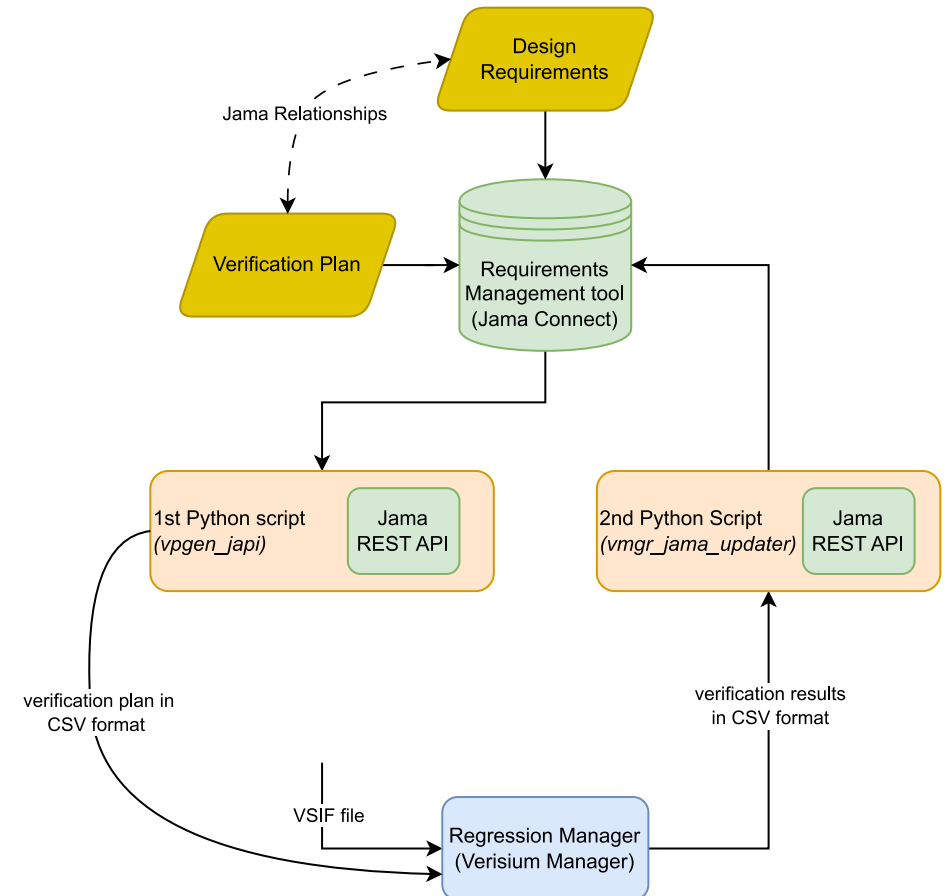
PROJECT ID:  
JK\_SB-TXT-289

GLOBAL ID:  
GID-370961

NAME:  
vsif\_cfg

DESCRIPTION:

Section	Name	Gates	RTL	Regression count	Regression timeout	Scan mode	Number of scan chains
section1	test1	no	yes	50	1		
section1	test2	yes	yes	10	1		
section2	test3	yes	no	10	1		
section2	test4	yes	yes	50	1		
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...



# Interface automation

## Regression session configuration:

Components / DVCON / Verification Plan / JK\_SB-TXT-289

vsif\_cfg V5

Text • Modified 10/23/2023 01:09:16 pm

Impact analysis

API ID:  
633636

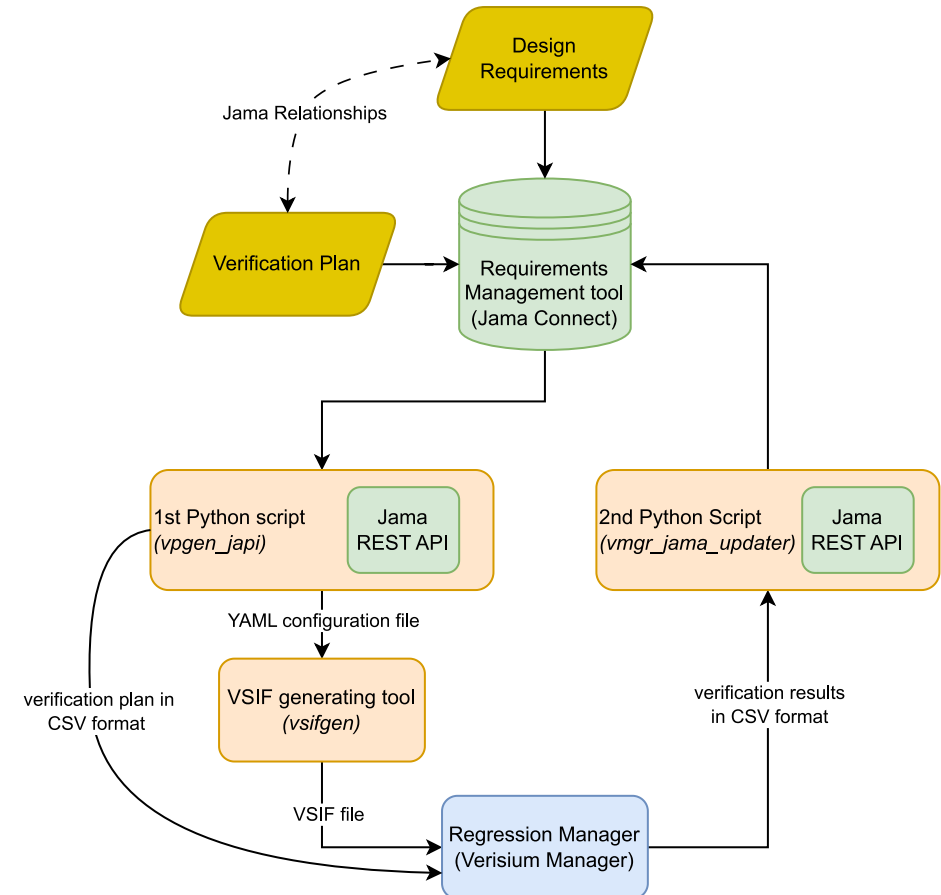
PROJECT ID:  
JK\_SB-TXT-289

GLOBAL ID:  
GID-370961

NAME:  
vsif\_cfg

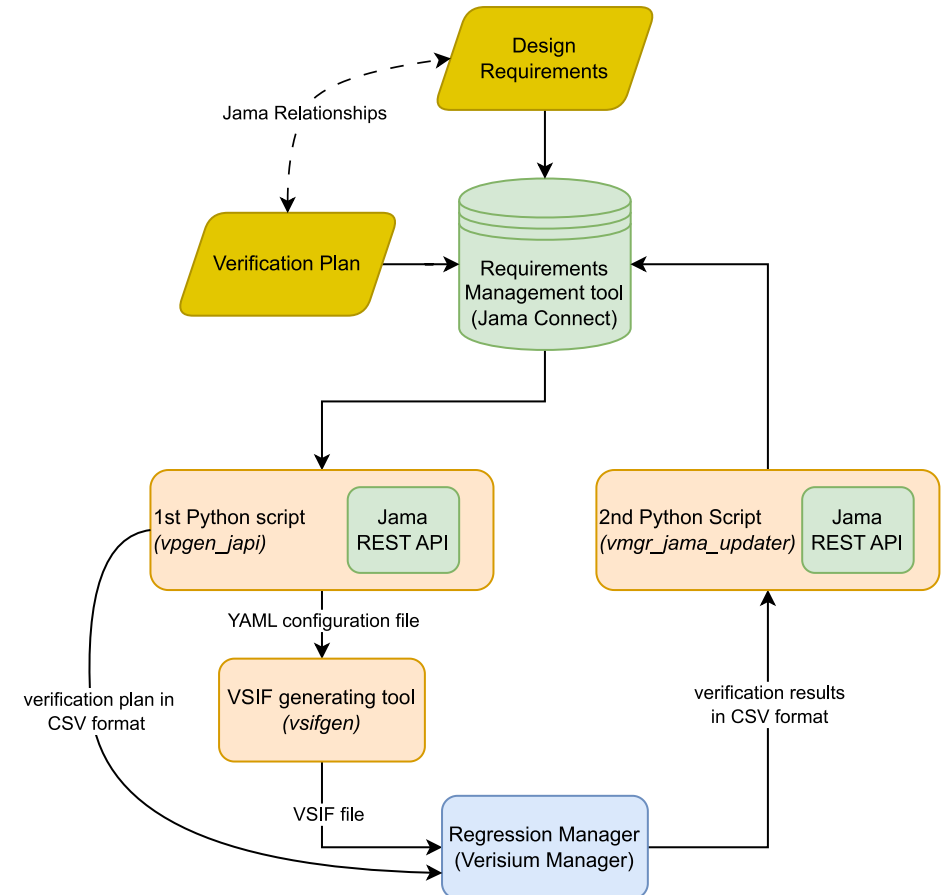
DESCRIPTION:

Section	Name	Gates	RTL	Regression count	Regression timeout	Scan mode	Number of scan chains
section1	test1	no	yes	50	1		
section1	test2	yes	yes	10	1		
section2	test3	yes	no	10	1		
section2	test4	yes	yes	50	1		
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...



# Interface automation

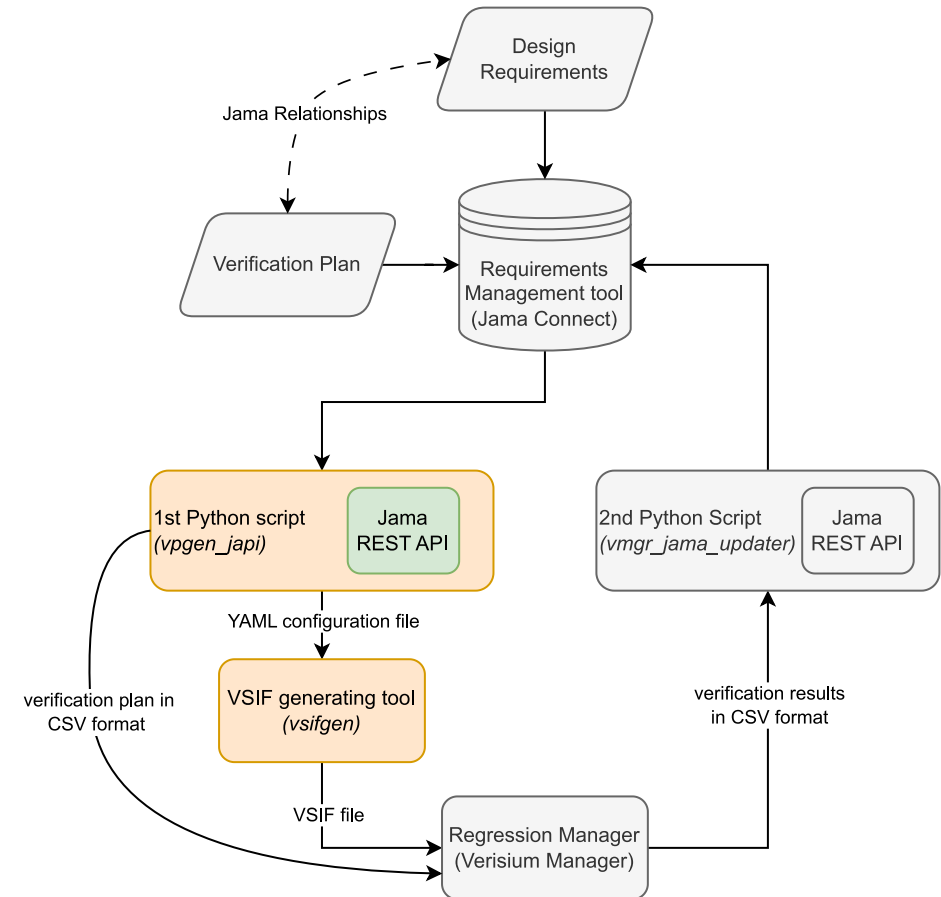
- Jama REST API
  - HTTP commands
  - Received data in JSON format
  - Open-source Python client
- Verisium Manager
  - batch mode commands





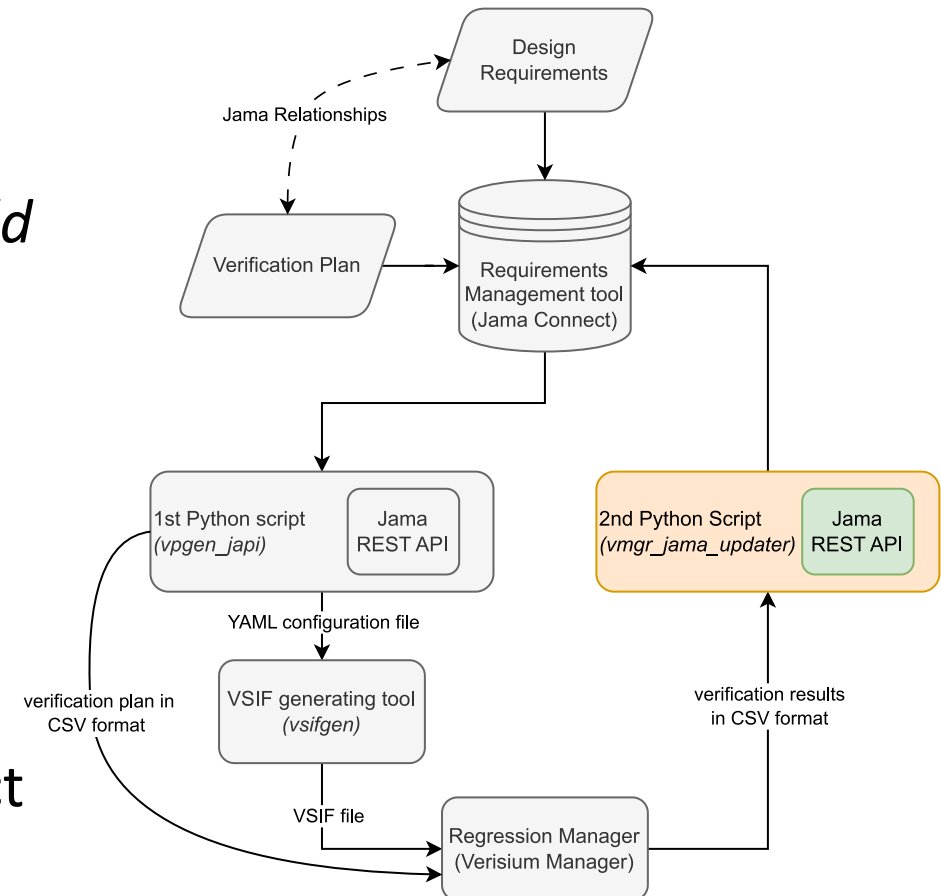
# From Jama Connect to Verisium Manager

- Vpgen\_japi script
  1. Authenticate API client
  2. Download all items within the project
  3. Filter out verification plan items
  4. Update or create VSIF config. item
  5. Generate CSV file
  6. Generate YAML files



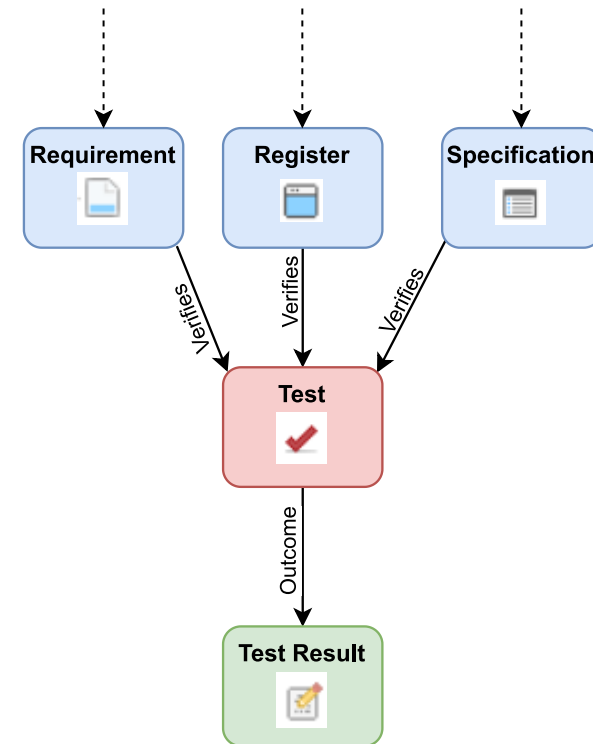
# Back to Jama Connect

- Vmgr\_jama\_updater script
    - Input argument: *ProjectID + V. Manager batch mode arguments + passing threshold*
1. Authenticate API client
  2. Download all items and relationships within the project
  3. Merge multiple regression sessions
  4. Apply the verification plan in CSV
  5. Generate regression report
  6. Upload regression report to Jama project



# Back to Jama Connect

- Vmgr\_jama\_updater script
    - Input argument: *ProjectID + V. Manager batch mode arguments + passing threshold*
1. Authenticate API client
  2. Download all items and relationships within the project
  3. Merge multiple regression sessions
  4. Apply the verification plan in CSV
  5. Generate regression report
  6. Upload regression report to Jama project
  7. Consistency check



TEST RESULT STATUS:\*

Not Run

Not Run

Fail

Pass

# Traceability in Jama Connect

Suspect relationship ...

... caused by requirement1 and register1

1 LEVEL UP					SOURCE ITEMS				
Component Requirement (V3) (2), Register (2), Specification (V3) (2)					Test (6), Text (1)				
Project ID	Name	Description	Relationship Status	Project ID	Name	Description	Relationship Status		
JK_SB-CREQ3-653	requirement1		?	JK_SB-TEST-446	testcase1	Brief brief description ...	!		
JK_SB-SPEC3-434	specification1		?	JK_SB-TEST-447	testcase2	Brief brief description ...	?		
JK_SB-REG-909	register1		?	JK_SB-TEST-448	assertion1	Brief brief description ...	!		
JK_SB-CREQ3-654	requirement2		?	JK_SB-TEST-449	coverage1	Brief brief description ...	!		
JK_SB-SPEC3-434	specification2		?						
JK_SB-REG-910	register2		?						
JK_SB-CREQ3-653	requirement1		?						
JK_SB-REG-909	register1		?						

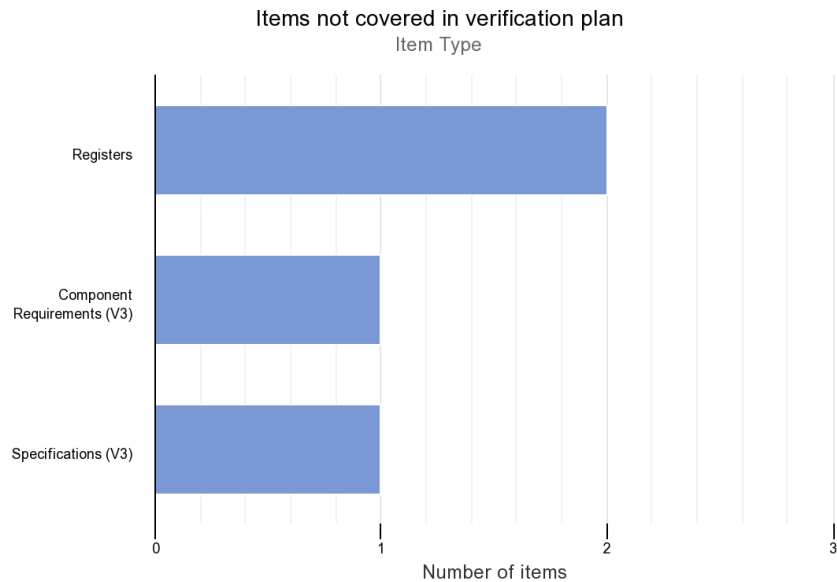
1 LEVEL UP					SOURCE ITEMS				
Test (6)					Test Result (6)				
Project ID	Name	Description	Relationship Status	Project ID	Name	Description	Relationship St...	Test Result Stat...	
JK_SB-TEST-446	testcase1	Brief brief description ...	!	JK_SB-RLT-1880	testcase1_result	TESTCASE: 0%	?	Not Run	
JK_SB-TEST-447	testcase2	Brief brief description ...	?	JK_SB-RLT-1884	testcase2_result	TESTCASE: 100%	?	Pass	
JK_SB-TEST-448	assertion1	Brief brief description ...	!	JK_SB-RLT-1885	assertion1_result	CHECK: 100%	?	Pass	
JK_SB-TEST-449	coverage1	Brief brief description ...	!	JK_SB-RLT-1886	coverage1_result	COVERAGE: 75%	?	Fail	
JK_SB-TEST-450	coverage2	Brief brief description ...	?	JK_SB-RLT-1887	coverage2_result	COVERAGE: 0%	?	Not Run	
JK_SB-TEST-451	testcase3	Brief brief description ...	?	JK_SB-RLT-1888	testcase3_result	TESTCASE: 95%	?	Fail	



# Traceability in Jama Connect

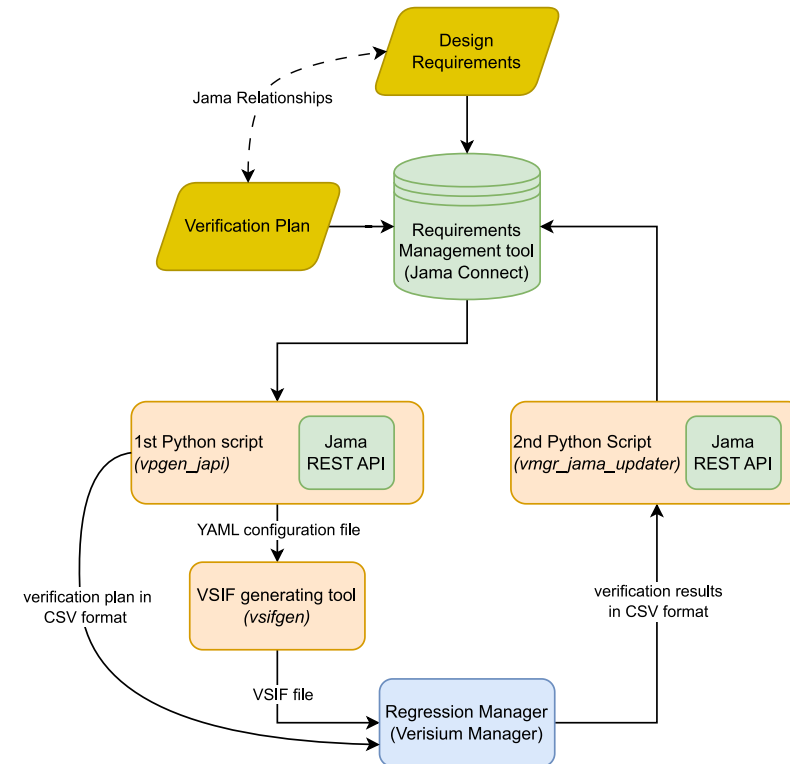
## Project status overview

- Regression status
- Verification plan completeness



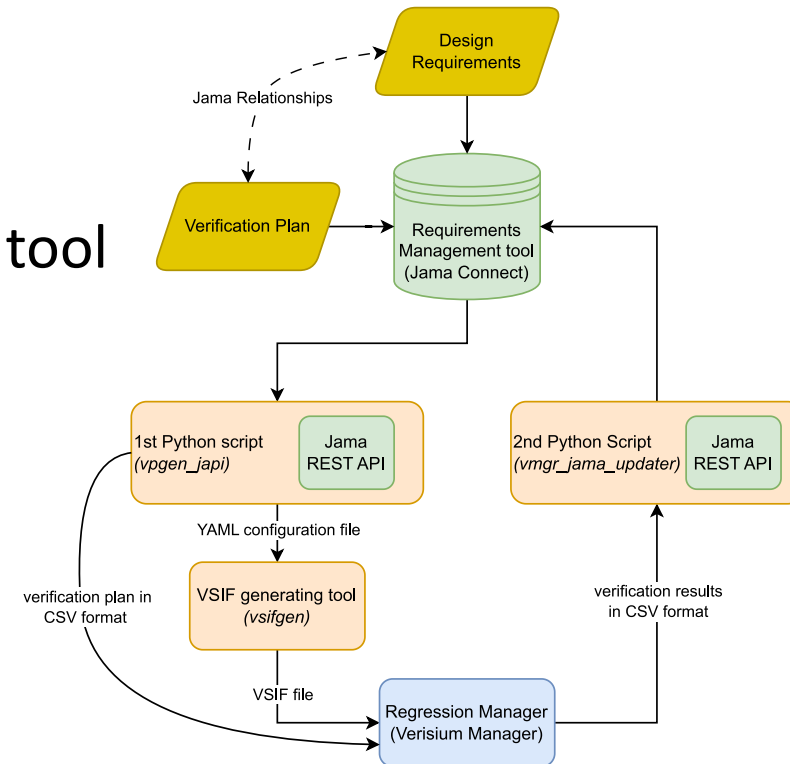
# Beyond Jama Connect and Verisium Manager

- Prerequisites:
  - API
  - Files with standard format
- RM tools:
  - Polarion (Siemens), Valispace, DOORS (IBM), ...
- Regression manager tools
  - Simulink (Mathworks), ...



# Summary

- Workflow used on several projects
- Expected benefits:
  - Verification results traceability in a single tool
  - Better project status overview
    - Single dashboard for DV, MSDV and Analog verification
  - Minimization of human errors
  - Shorter execution cycles
  - Increase in verification results quality
  - License usage optimization





# Questions

Jan Kreisinger, Sanjay Chatterjee

[jkreisinger@allegromicro.com](mailto:jkreisinger@allegromicro.com), [schatterjee@allegromicro.com](mailto:schatterjee@allegromicro.com)

Allegro MicroSystems



# Test item in JSON format

verification\_element\_name V2 ▾

✓ Test • Modified 10/19/2023 12:50:34 pm

▾ Impact analysis



PROJECT ID:

[JK\\_SB-TEST-446](#)

GLOBAL ID:

[GID-394678](#)

NAME:

verification\_element\_name

DESCRIPTION:

<b>Brief</b>	<i>brief description</i>
<b>Detailed</b>	<i>detailed description</i>
<b>Section</b>	<i>section name</i>
<b>Metrics port kind</b>	TESTCASE or COVERAGE or CHECK
<b>Mapping pattern</b>	default

Jama REST API - GET



```
{
  'id': 123456,
  'documentKey': 'JK_SB-TEST-446',
  'itemType': 1,
  'fields': {
    'documentKey': 'JK_SB-TEST-446',
    'globalId': 'GID-394678',
    'name': 'verification_element_name',
    'description': '<table>...</table>\n',
  },
  'resources': {
    'self': {
      'allowed': ['GET', 'PUT', 'PATCH', 'DELETE']}},
  'location': {
    'parent': {
      'item': 123455}},
  'type': 'items'
}
```