

# Smarter Verification Management with vManager Platform

David Zhang, AE Director, Cadence



# Agenda



Fundamentals of Smarter Verification Management

Next-Generation Architecture

Centralized Multi-Geography Collaboration

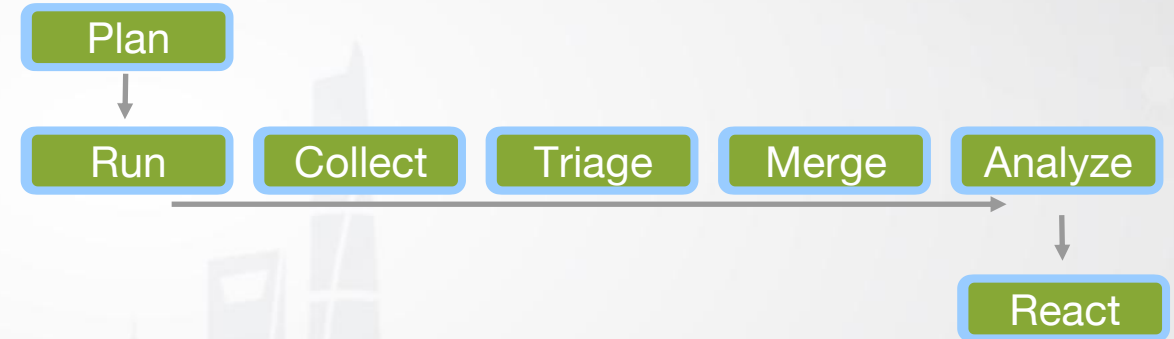
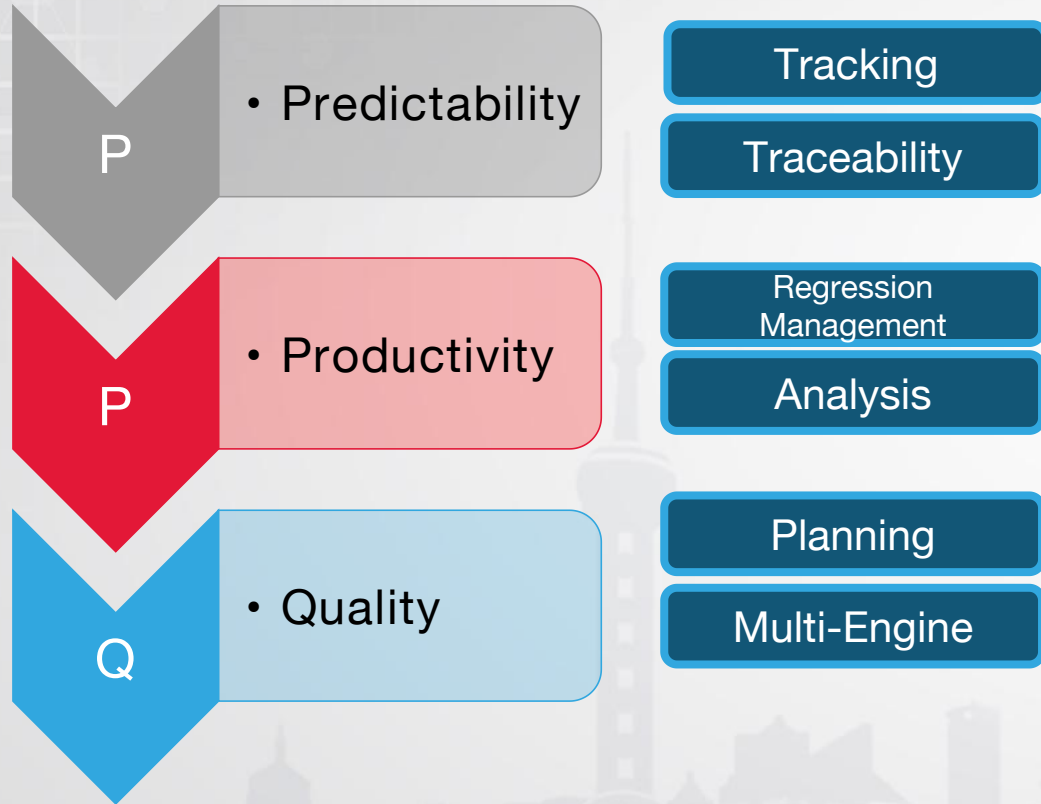
Multi-User, Multi-Engine Concurrent Verification Planning

Automation and Customization

# Fundamentals of Smart Verification Management

It's all about predictability, productivity and quality

# Better PPQ Verification Management Across the Flow



# Productivity: Regression Round Trip

The screenshot displays the Cadence vManager interface for analyzing test results. The main window is titled "vManager radical:8080 (64) [Analysis - Messages Analysis]". The interface includes a menu bar (File, View, Analysis, Help) and a toolbar with various analysis tools. The "Analysis" tab is active, showing a "Groups of Errors" panel on the left and a "Runs" panel on the right.

**Groups of Errors Panel:**

Name	Description	Number Of Entit
(no filter)	(no filter)	(no filter)
SCRB	##### FAIL : APB RECEIVED WRONG DATA from uart0	28
Error	Parity mismatch_20	6
Error	Parity mismatch_19	5
Error	Unexpect CACHE miss_8	5
Error	Attempted access to illegal memory location_19	4
Error	Unexpect CACHE miss_14	4
Error	Parity mismatch_9	4
Error	Low speed channel SB error_7	4
Error	RIFO write ptr out of bounds_1	4
Error	Parity mismatch_7	4
Error	Low speed channel SB error_1	4

Showing 111 items

**Runs Panel:**

Index	Name	Status	First Failure Time	Seed	Run
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
9	/uart_tests/apb_uart_rx_tx	failed	213500	165...	/ho...
18	/uart_tests/uart_apb_incr_data	failed	1080500	-13...	/ho...
10	/uart_tests/apb_uart_rx_tx	failed	1751500	-16...	/ho...
12	/uart_tests/apb_uart_rx_tx	failed	1920500	212...	/ho...
17	/uart_tests/uart_apb_incr_data	failed	1977500	100...	/ho...
14	/uart_tests/uart_apb_incr_data	failed	2016500	633...	/ho...
8	/uart_tests/apb_uart_rx_tx	failed	2480500	128...	/ho...
16	/uart_tests/uart_apb_incr_data	failed	2804500	-18...	/ho...

Showing 8 items

**Details Panel:**

Attributes Logs

brun.log x xrun\_apb\_uart\_rx\_tx.log x local\_log.log x local\_log.log x

338 UVM\_INFO /home/magraham/xlm\_workshop/sv\_cb\_ex\_lib/uart\_ctrl/sv/sequence\_lib/...  
339 UVM\_INFO /home/magraham/xlm\_workshop/sv\_cb\_ex\_lib/interface\_uvc\_lib/uart/sv/u...  
340 UVM\_INFO /home/magraham/xlm\_workshop/sv\_cb\_ex\_lib/uart\_ctrl/sv/sequence\_lib/i...  
341 UVM\_ERROR /home/magraham/xlm\_workshop/sv\_cb\_ex\_lib/uart\_ctrl/sv/uart\_ctrl\_sco...  
342 UVM\_INFO /home/magraham/xlm\_workshop/sv\_cb\_ex\_lib/uart\_ctrl/sv/uart\_ctrl\_sco...  
343 UVM\_ERROR /home/magraham/xlm\_workshop/sv\_cb\_ex\_lib/uart\_ctrl/sv/uart\_ctrl\_sco...  
344 UVM\_INFO /home/magraham/xlm\_workshop/sv\_cb\_ex\_lib/uart\_ctrl/sv/uart\_ctrl\_sco...  
345 UVM INFO /arid/avs/install/xcelium/AGILE/18.04.001/tools/methodoloav/UVM/CDNS...

Showing 28 items

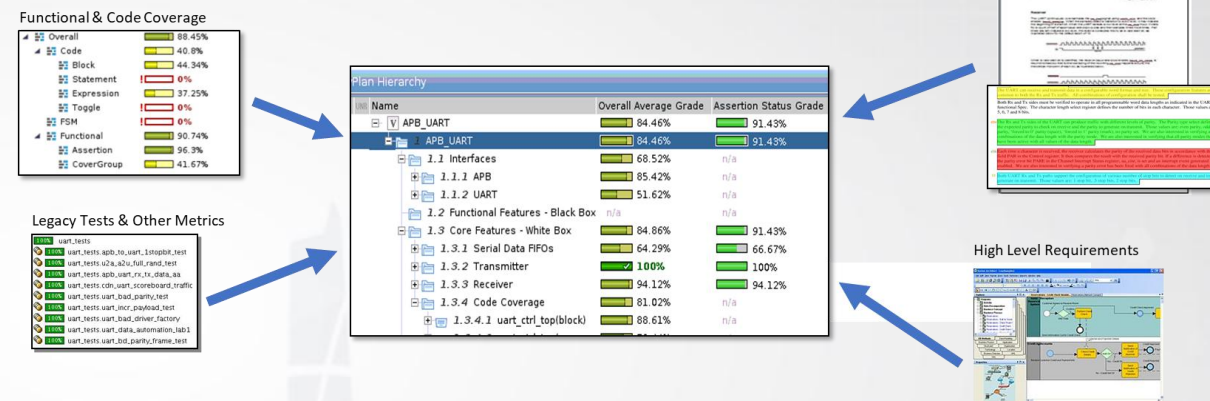
Loaded Run: Session

Messages

- Prioritize highest impact failure modes
- Identify shortest test for each failure mode
- Automated rerun to generate debug data

# Predictability: Traceability

- Requirements traceability
  - Requirement import to seed verification plan creation
  - Link requirements and verification plan
  - Visibility of changes to requirements



- Coverage analysis
  - Multi-engine coverage merge/combine
  - Refinement (unreachability (UNR), UNR crosses)
  - Analysis of coverage vs. plan

The screenshot displays the Synopsys Design Compiler (DC) interface with three main panes:

- Instances and Types:** Lists components like `i_ahbSlaveMonitor`, `i_apb_lite`, `i_ahb_lite`, and `i_counter_lite`. Below this, the **Top Level Expressions** pane shows a table of expressions with their overall average grade, overall uncovered status, and source line.
- Source:** Displays the Verilog code for `pb_subsystem_top_for_smc.v`. The code includes a conditional statement for `htrns` and a `case` statement for `hresp`.
- Coverage Table with Output:** Shows a table of coverage data for the selected source file. The table includes columns for index, T1, T3, T4, T6, T7, T8, T9, T10, and Score.

**Top Level Expressions Table:**

Index	Overall Average Grade	Overall Uncovered	Source Line
3	100%	0	274
4	33.33%	2	283
5	50%	1	292
6	100%	0	332
7	36.36%	7	338

**Coverage Table with Output:**

Index	T1	T3	T4	T6	T7	T8	T9	T10	Score
4	0	-	1	1	0	0	-	-	1
5	-	-	-	-	1	0	1	0	0
6	-	-	-	-	1	-	0	1	0
7	-	-	0	0	-	-	0	1	1
8	-	-	0	-	-	0	1	1	1

# Quality: Defined and Agreed Goals and Metrics for Entire Team

Identify  
Tests

Verification Plan

Link  
Specs

Plan Hierarchy			
UNB	Name	Overall Average Grade	Assertion Status Grade
[-]	APB_UART	84.46%	91.43%
[+]	1 APB_UART	84.46%	91.43%
[-]	1.1 Interfaces	68.52%	n/a
[+]	1.1.1 APB	85.42%	n/a
[+]	1.1.2 UART	51.62%	n/a
[-]	1.2 Functional Features - Black Box	n/a	n/a
[-]	1.3 Core Features - White Box	84.86%	91.43%
[+]	1.3.1 Serial Data FIFOs	64.29%	66.67%
[+]	1.3.2 Transmitter	100%	100%
[+]	1.3.3 Receiver	94.12%	94.12%
[-]	1.3.4 Code Coverage	81.02%	n/a
[+]	1.3.4.1 uart_ctrl_top(block)	88.61%	n/a

Identify  
Coverage

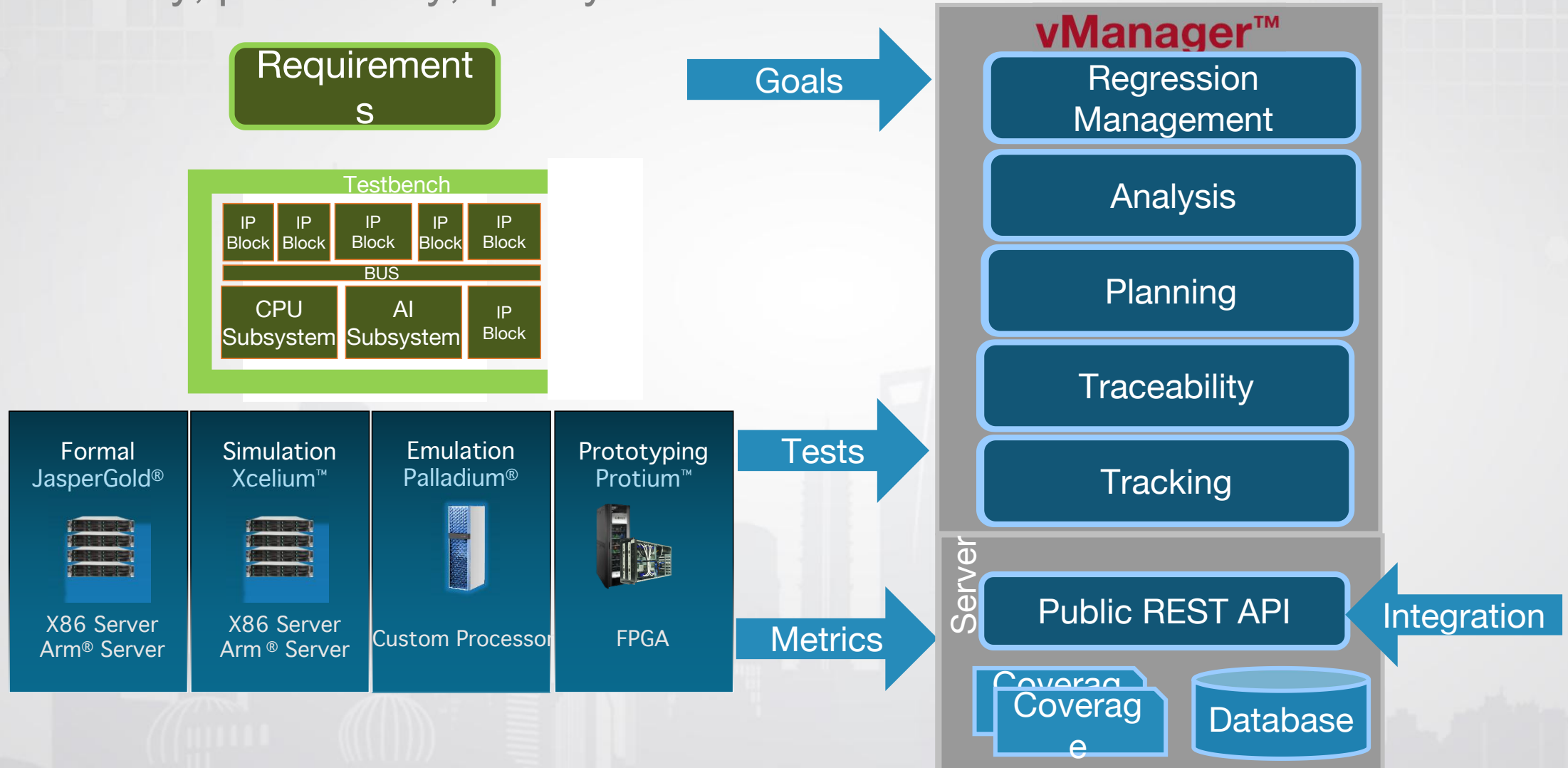
Import  
vPlans



Common Understanding

# Smart Verification Management

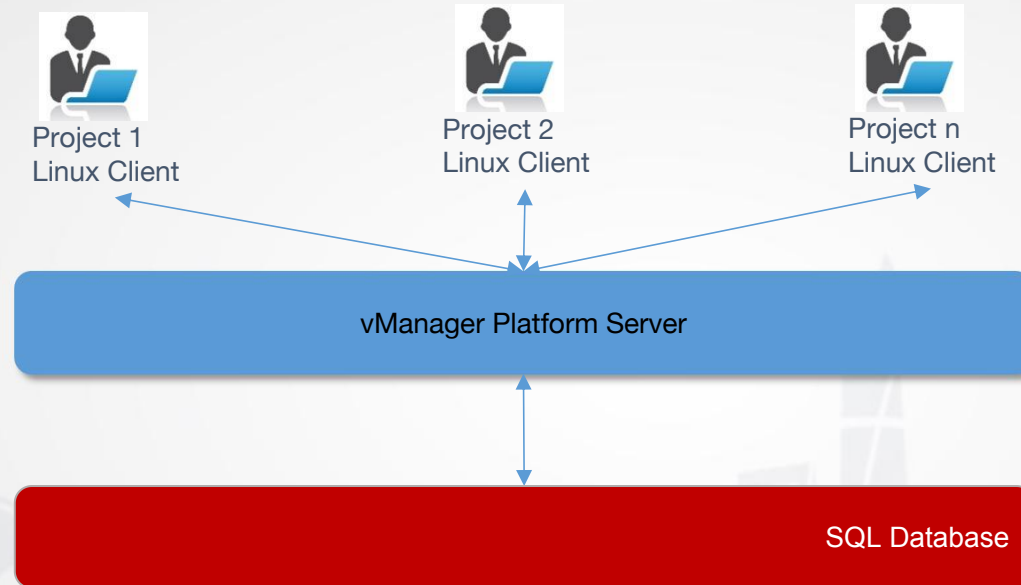
Predictability, productivity, quality



# Verification Management System Architecture

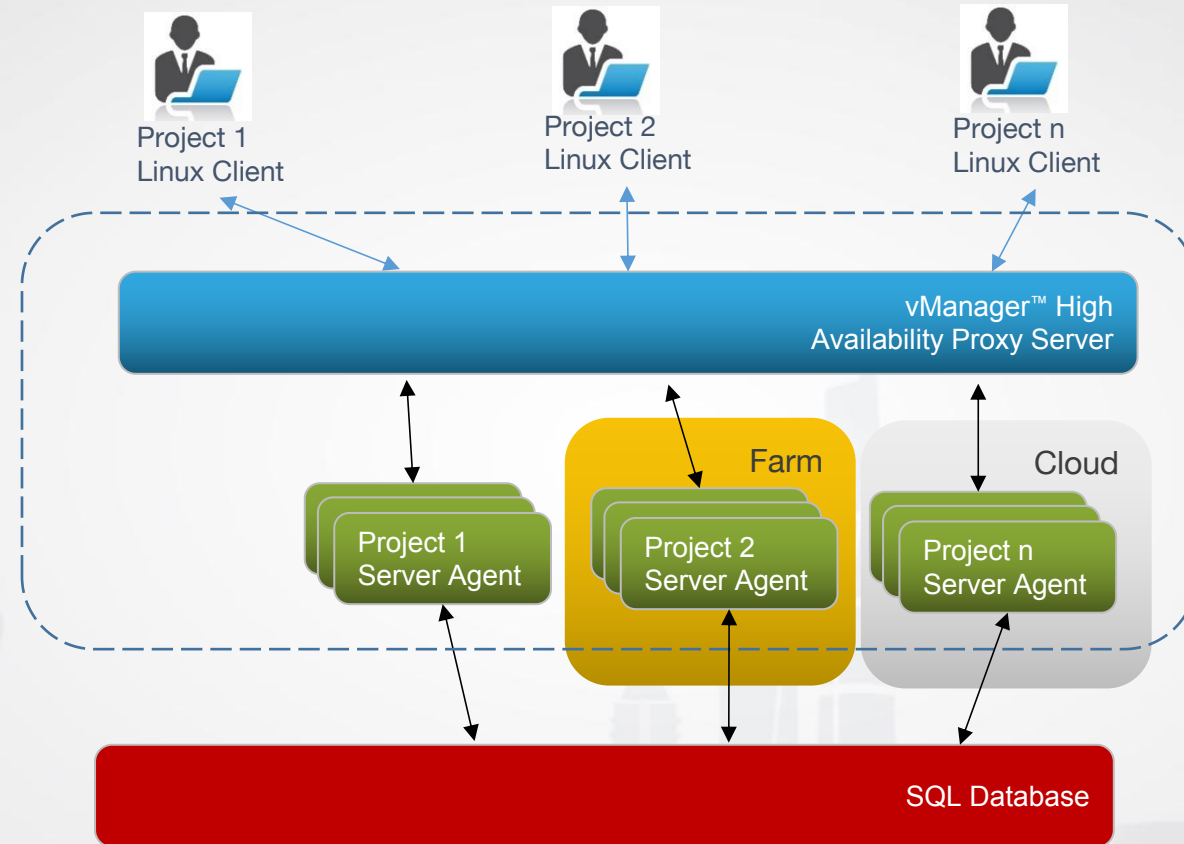
Scalability, reliability, and ease of admin

# Current vManager Architecture



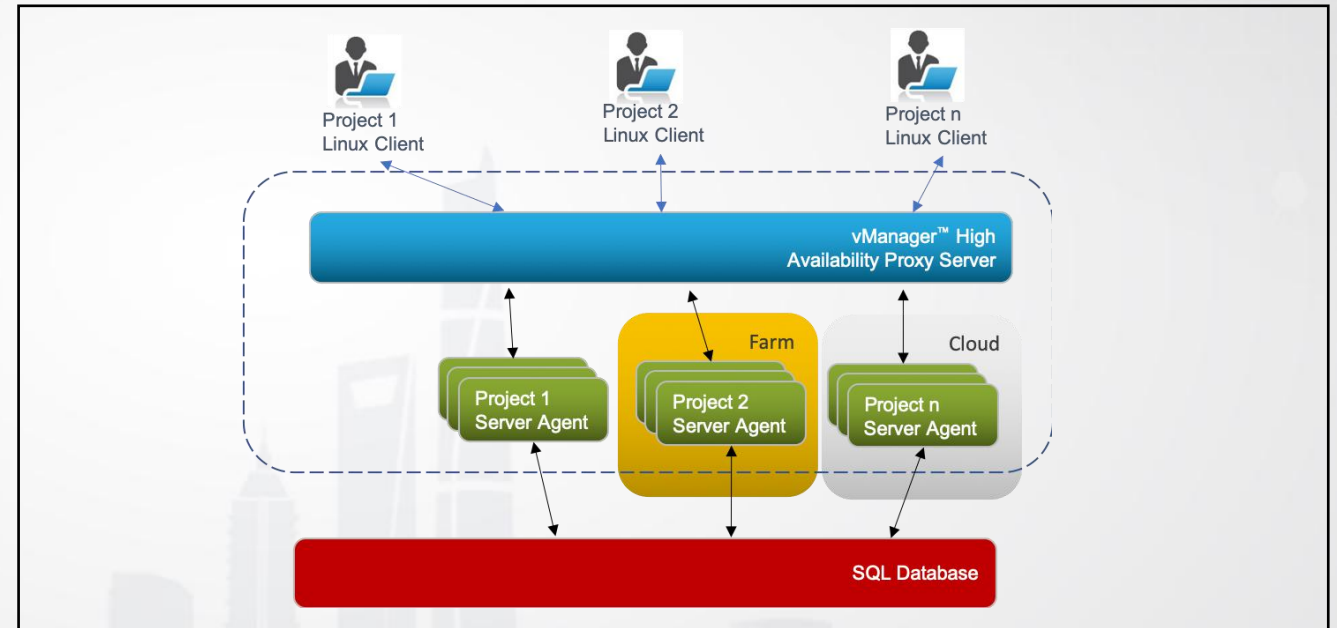
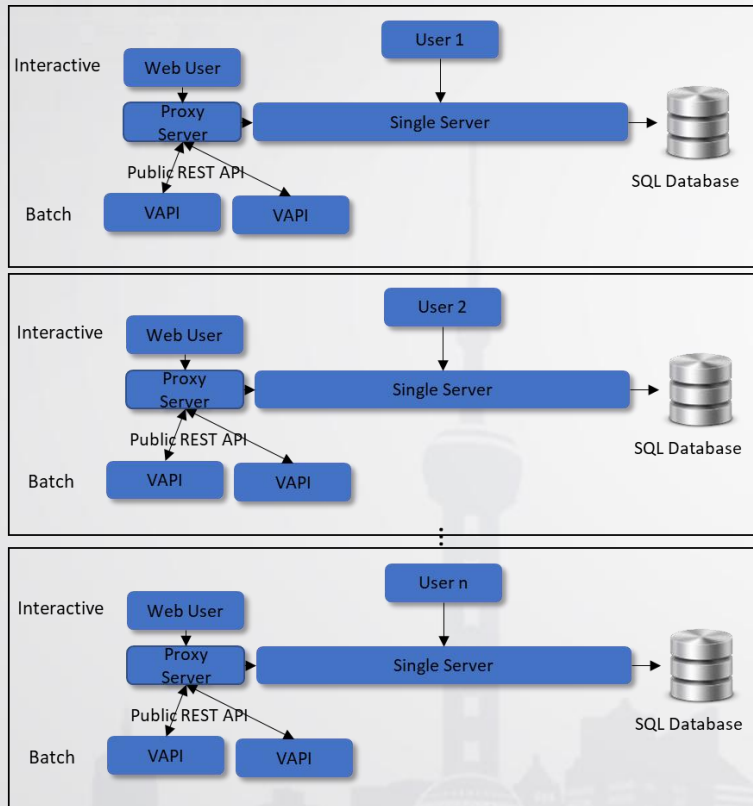
- Minimum requirement for a verification management system
  - Multi-user
  - Centralized database
  - Multiple project support

# Next-Generation Architecture



- Load balancing and resilient routing
- Distributed processing for redundancy, scalability, and fault tolerance
- Provide scalable, resilient, reliable replacement for previous single, monolithic system

# Distributed Architecture = Consolidation Opportunity



- Reduced maintenance overhead
- Lower IT infrastructure requirements
- Improved reliability and scale

# Multi-Geography Verification Management

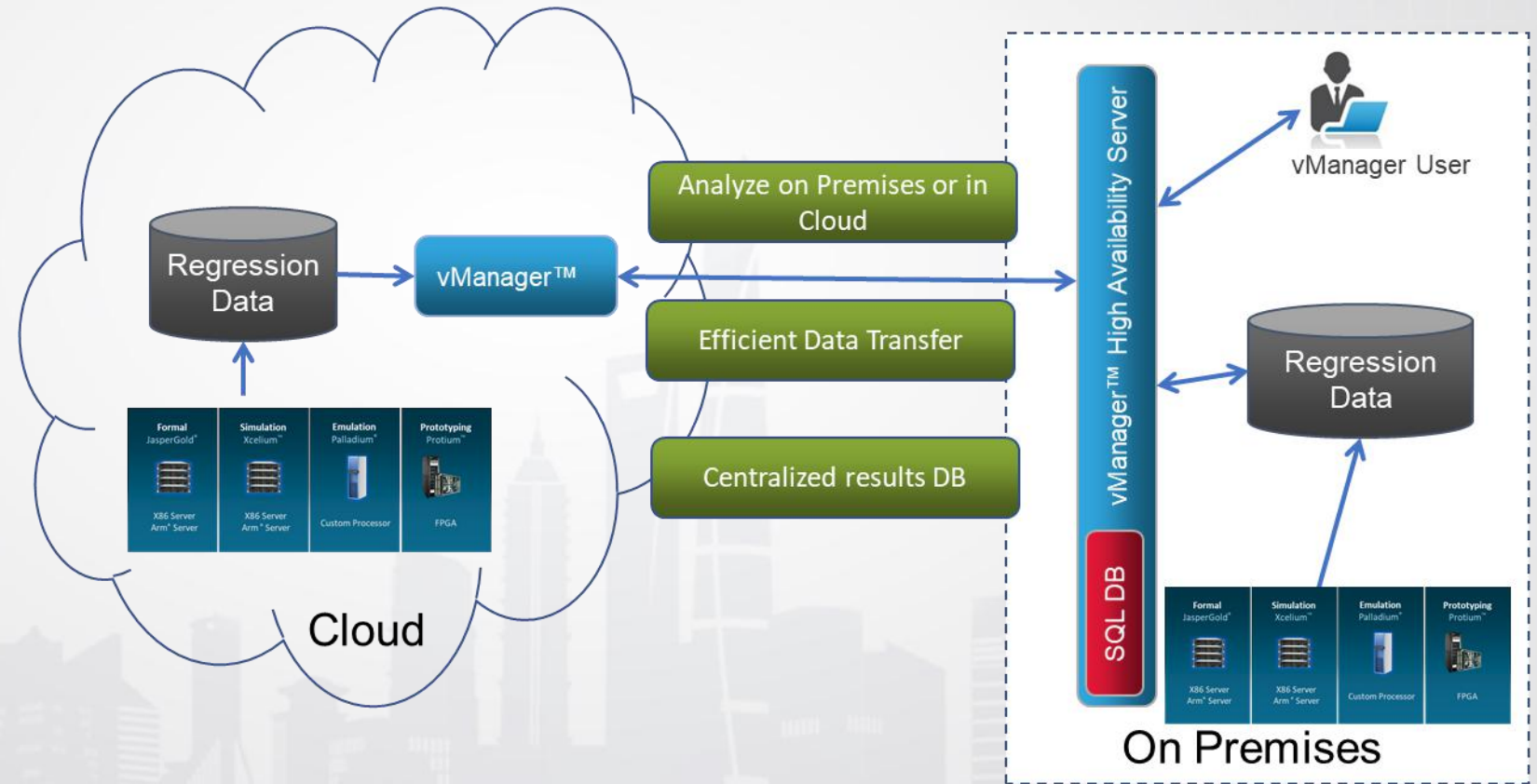
Utilizing the advanced architecture to enable data centralization

# Review – Basic Summary Data Sharing



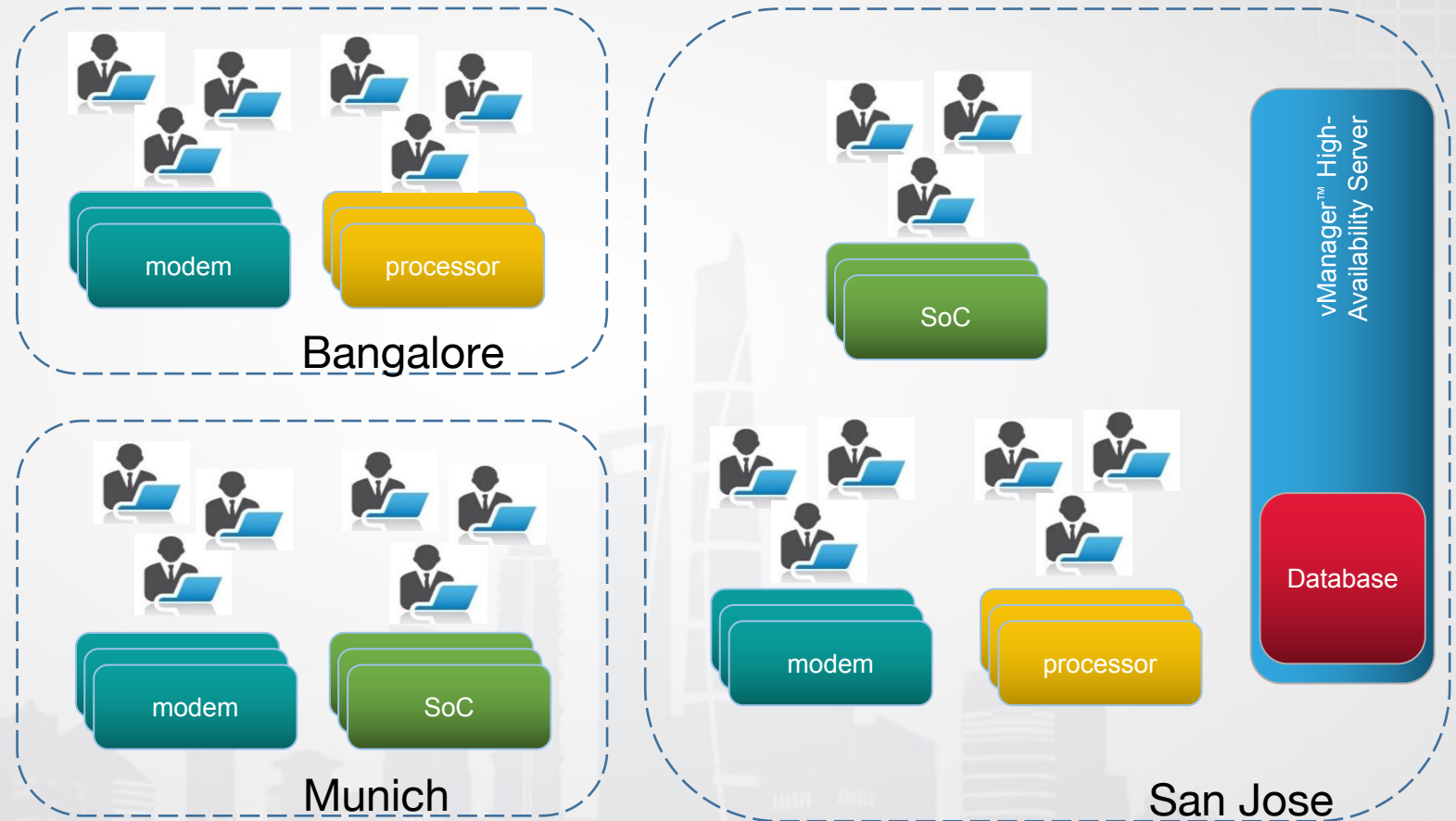
# True Multi-Region Capability

- Distributed, enterprise ready architecture
- Scale across multiple sites and into cloud
- Centralized regression database for collaboration



# Multi-Project, Multi-Region Topology

- Single vManager Server
- Central data management across projects
- Aggregated analysis and reporting across geographies



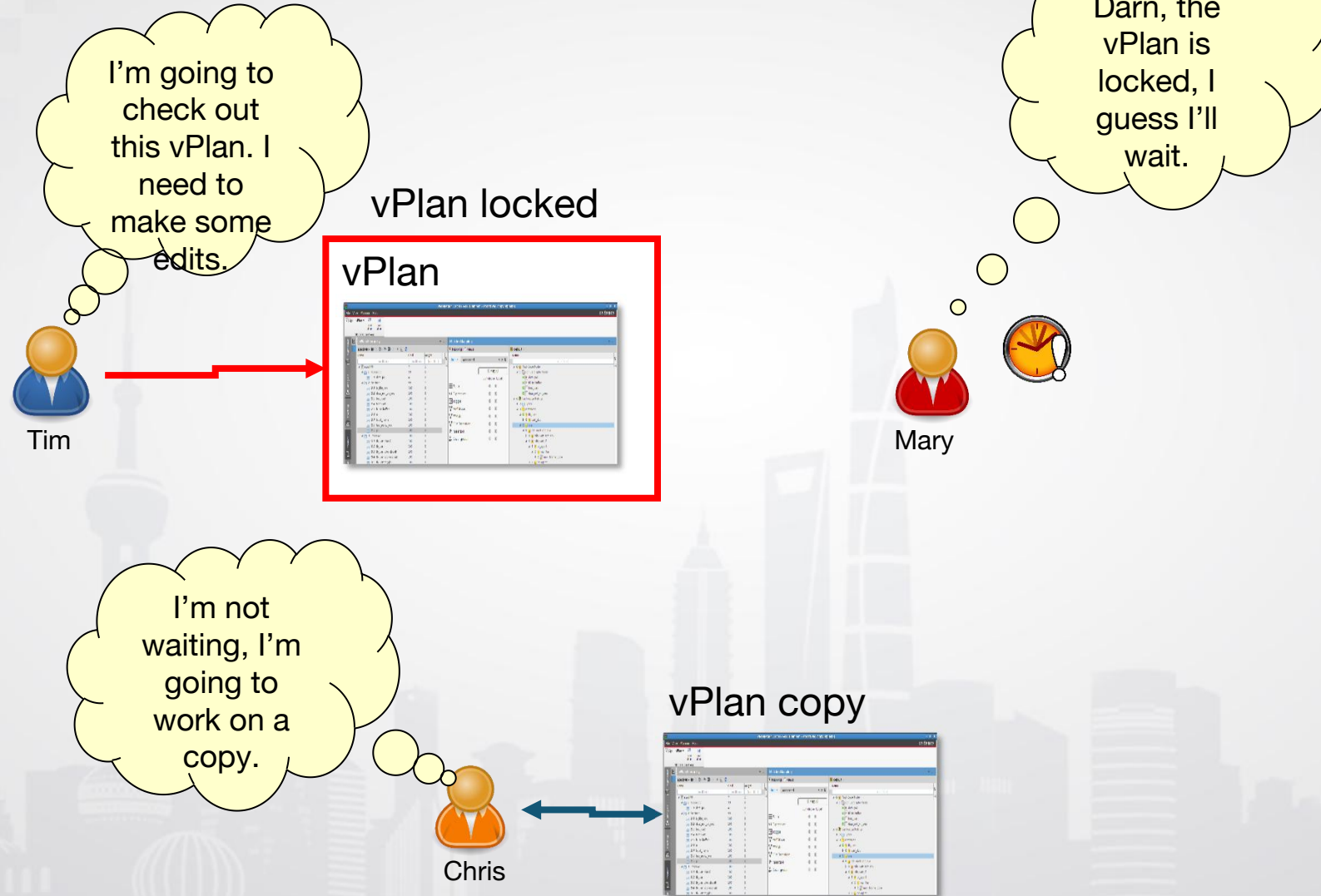
# Case Study: vManager实际部署及系统性能

- 项目部署数量: 15+
- vManager server 部署规模数量: 10+
- vPlan更新性能: 实时
- vAPI运行稳定: 全年零故障
- 看板提取数据性能
  - 和实际项目规模强相关
  - 完成一个项目的数据提取需约0.5~1小时

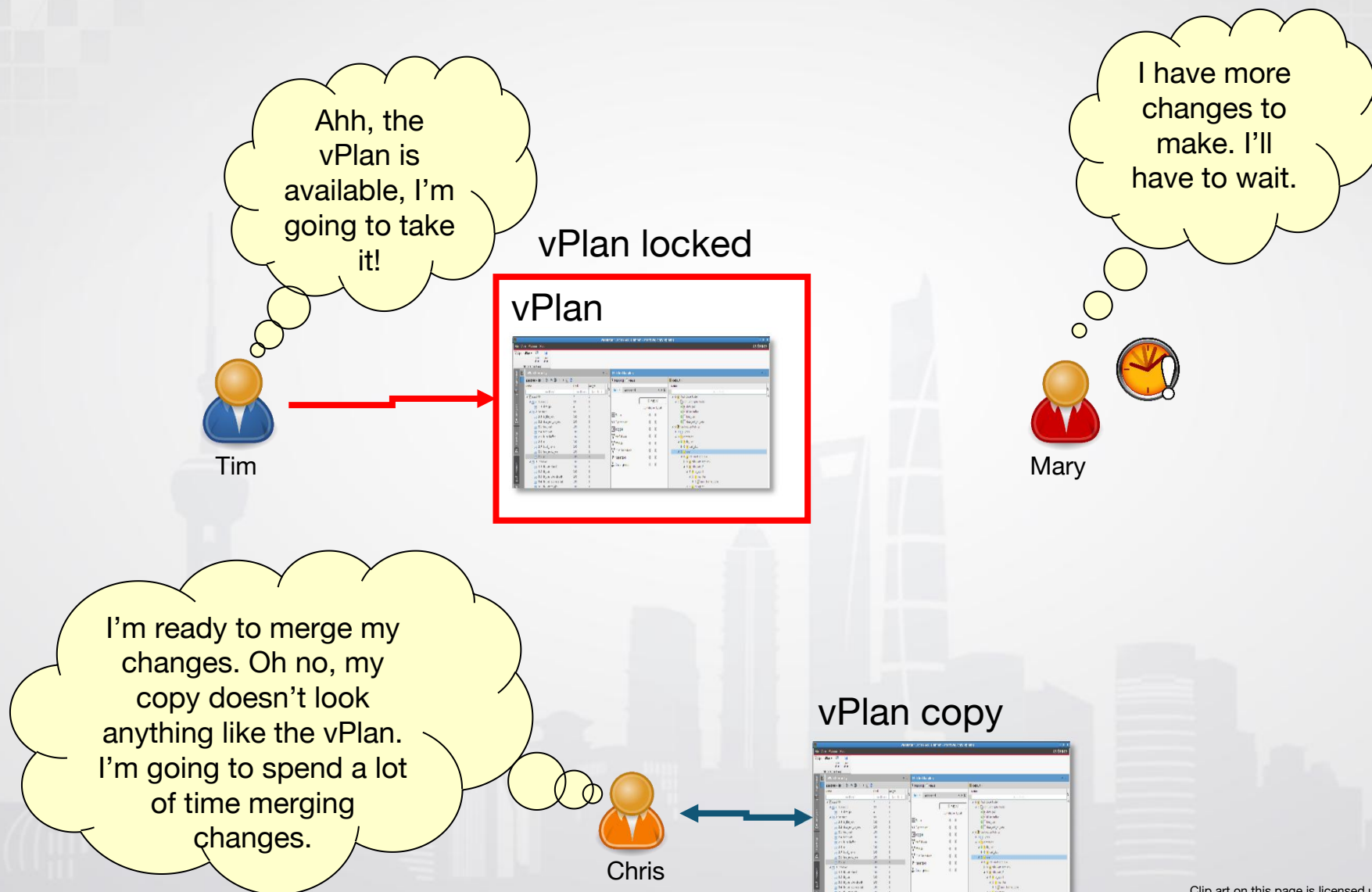
# Multi-User, Multi-Engine, Multi-Region Verification Planning

Providing collaboration along with advanced planning features

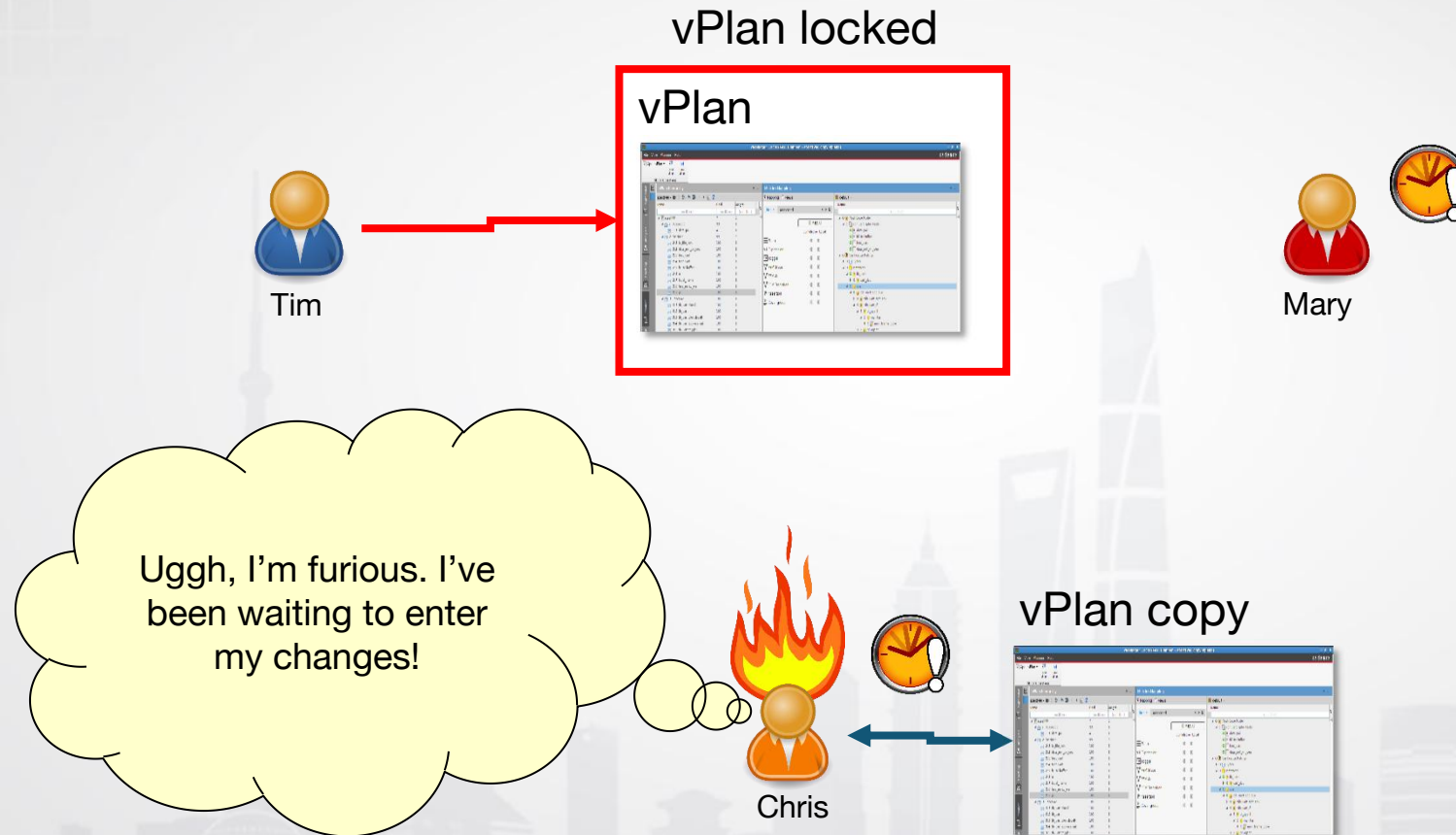
# File-Based Verification Plans



# File-Based Verification Plans

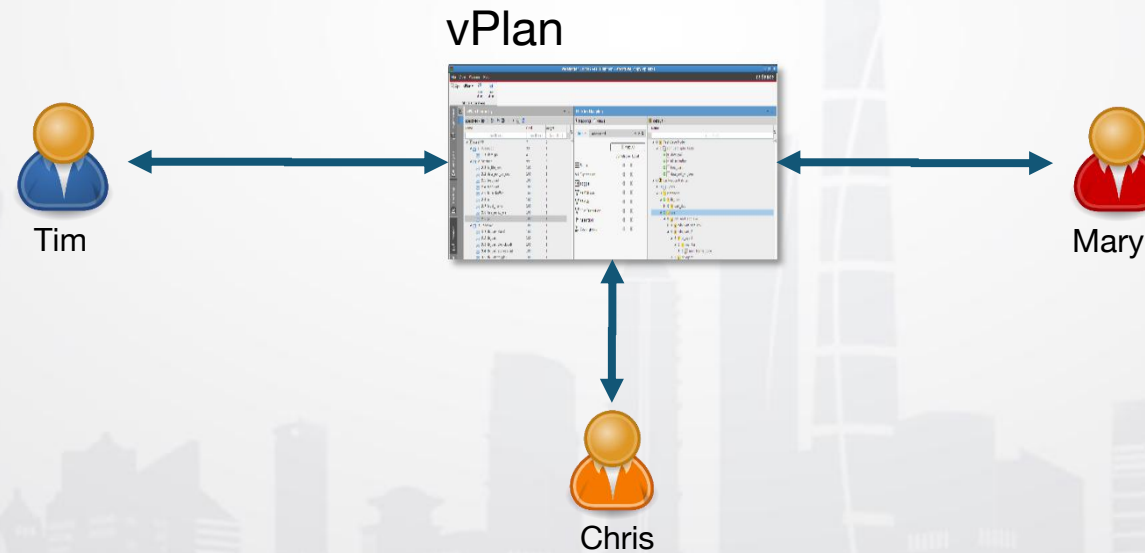


# File-Based Verification Plans

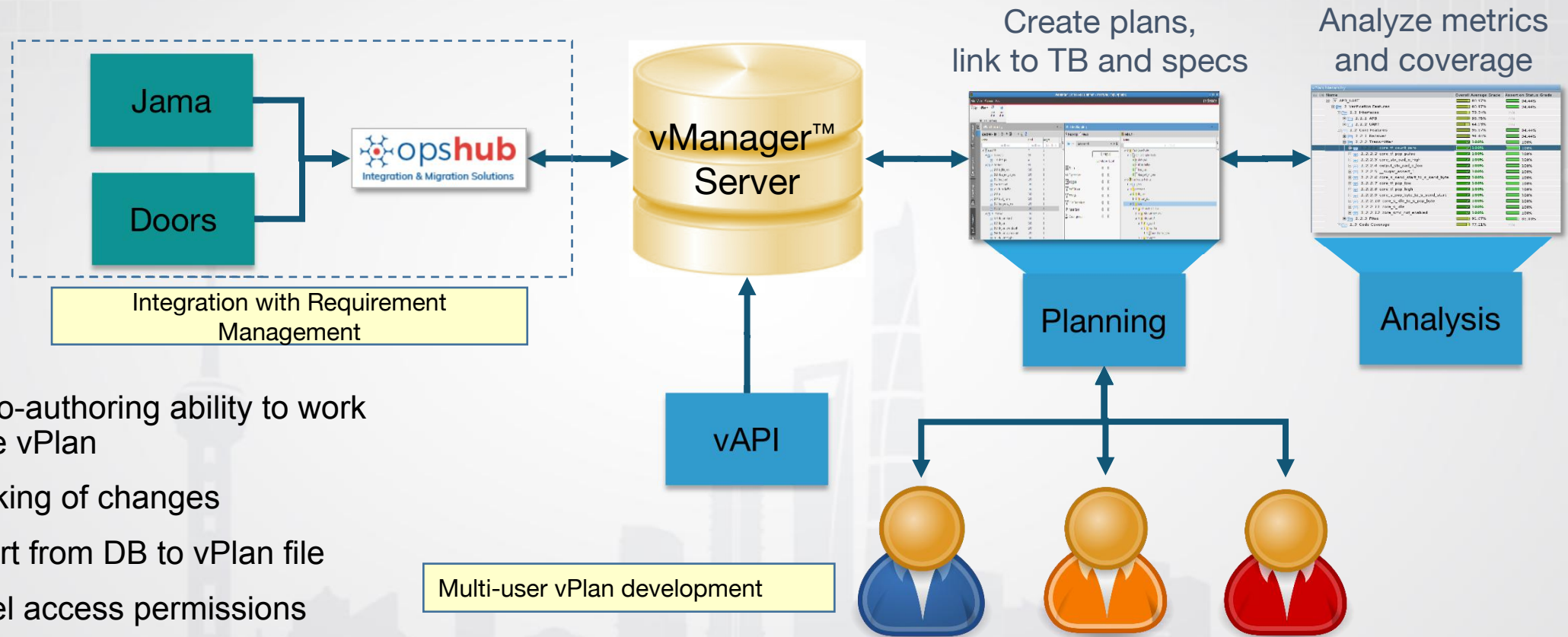


# Co-Authoring Verification Plans

- Move verification plan storage to the centralized database
- Enable co-authoring for multi-user, multi-region verification plan creation



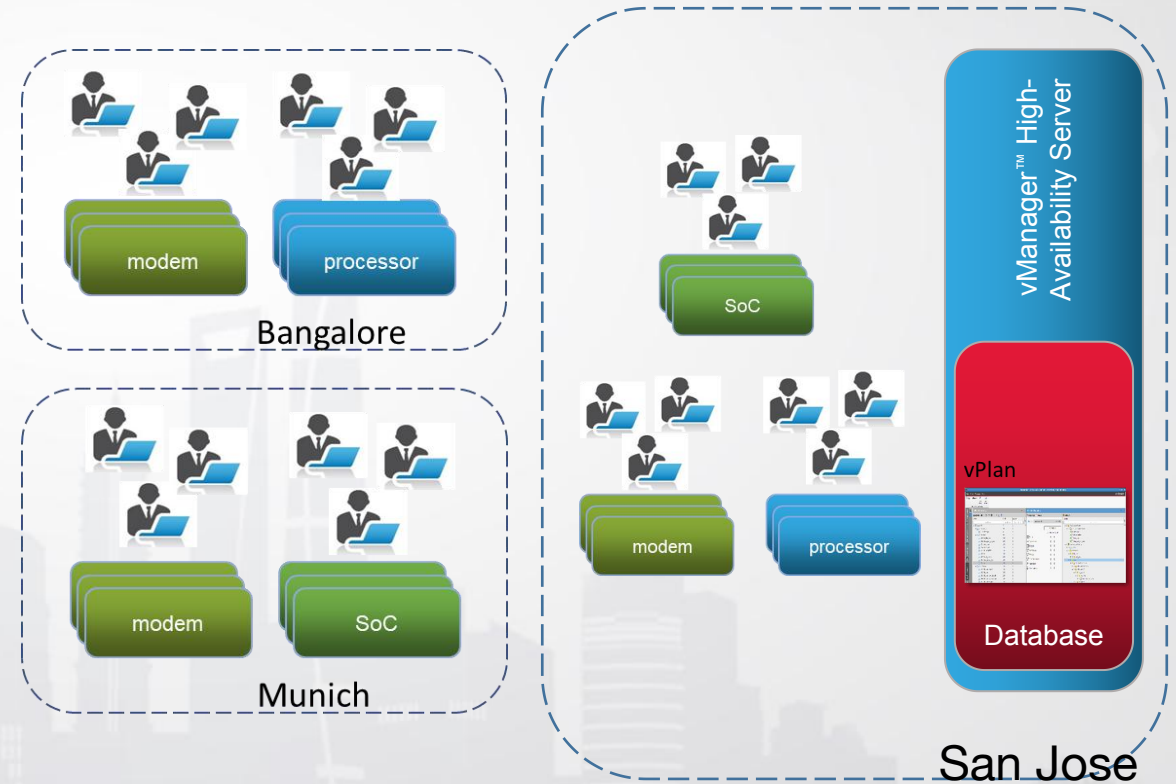
# Verification Plan in Database



- Multi-user co-authoring ability to work on the same vPlan
- History tracking of changes
- Import/export from DB to vPlan file
- Section-level access permissions

# Multi-Region vPlan Collaboration

- Single vPlan acts as Verification Contract for diverse teams
- Automatically in sync – no need for problematic cross-site document merge
- Central persistence for connection to external systems (e.g., req. mgmt.



# Case Study: Microsoft TFS 与vManager的数据交互



- TFS负责市场需求和产品需求的管理和跟踪
  - 提供源代码管理、数据收集、报告和项目跟踪
- TFS里的产品需求点能够自动转换成vPlan的验证需求条目
- TFS里的验证需求点产生变更，则vPlan的内容需要进行迭代更新，特别地，需要支持产品需求点被删除，取消，挂起，新建四种状态
- 验证需求完成情况自动翻转TFS中产品需求的完成状态

# Multi-Engine Case Study: Mobile SoC

Interconnect Workbench

Xcelium™ with UVM

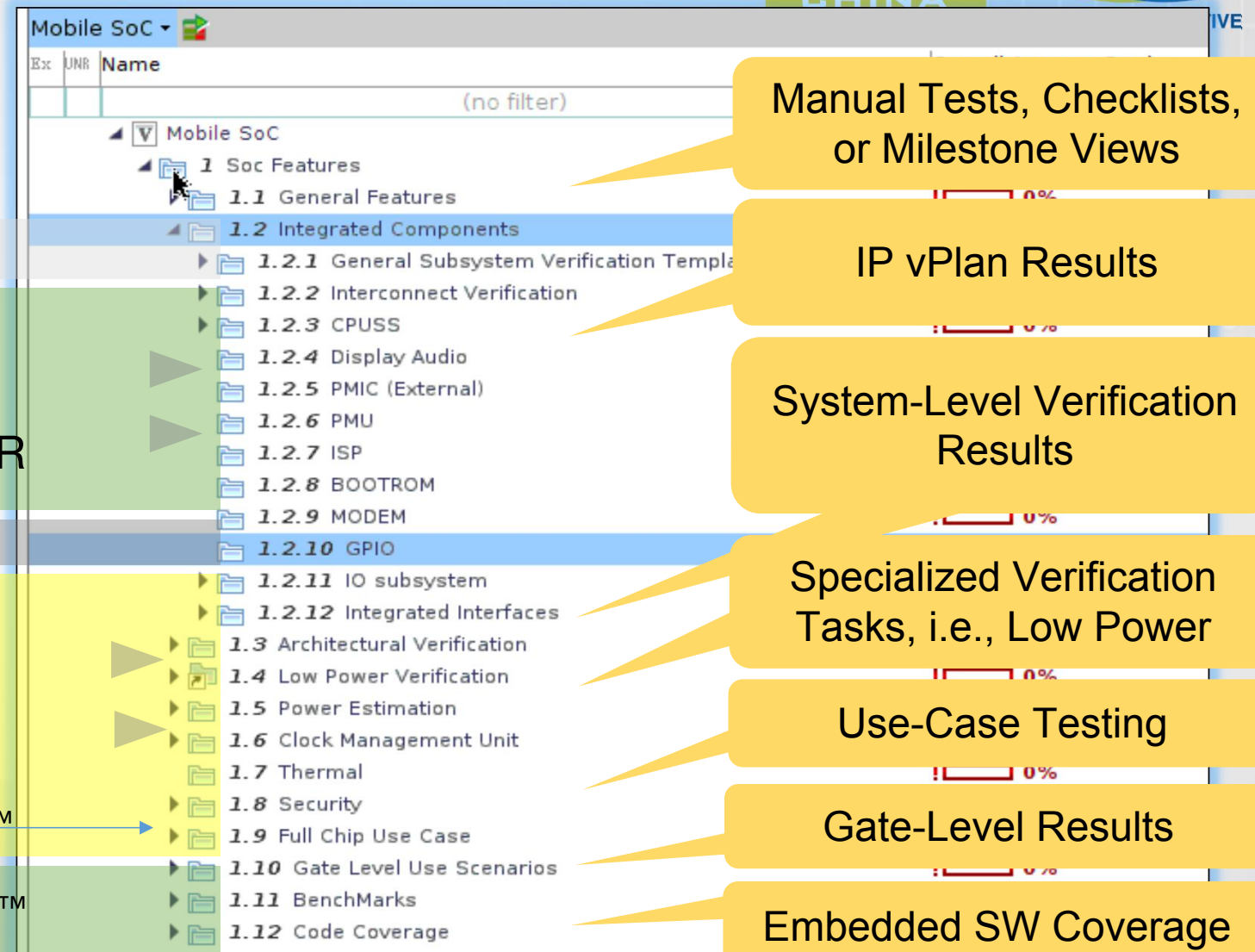
JasperGold® UNR

JasperGold Formal

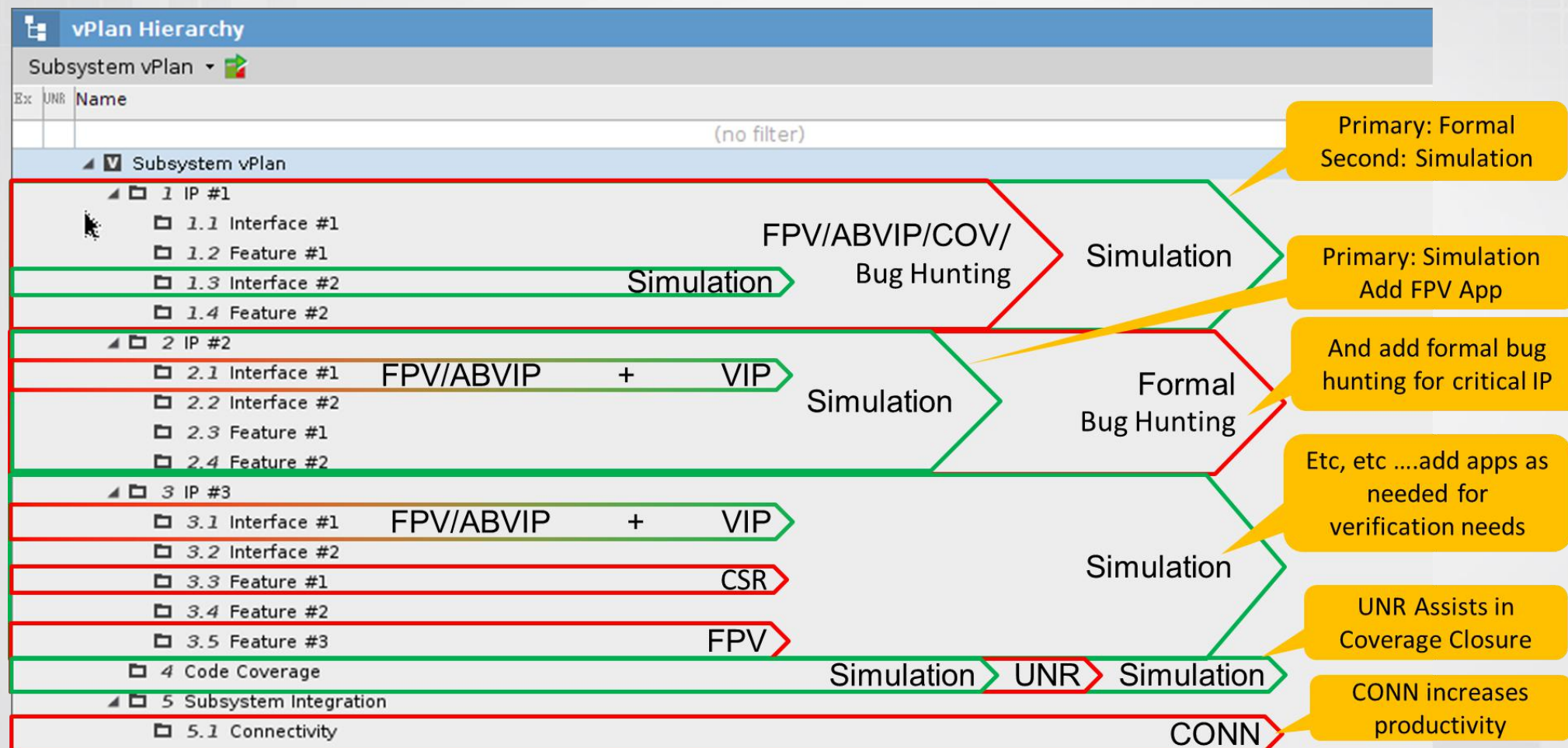
Palladium®

Perspec™

Protium™



# Sim + Formal Combined vPlan

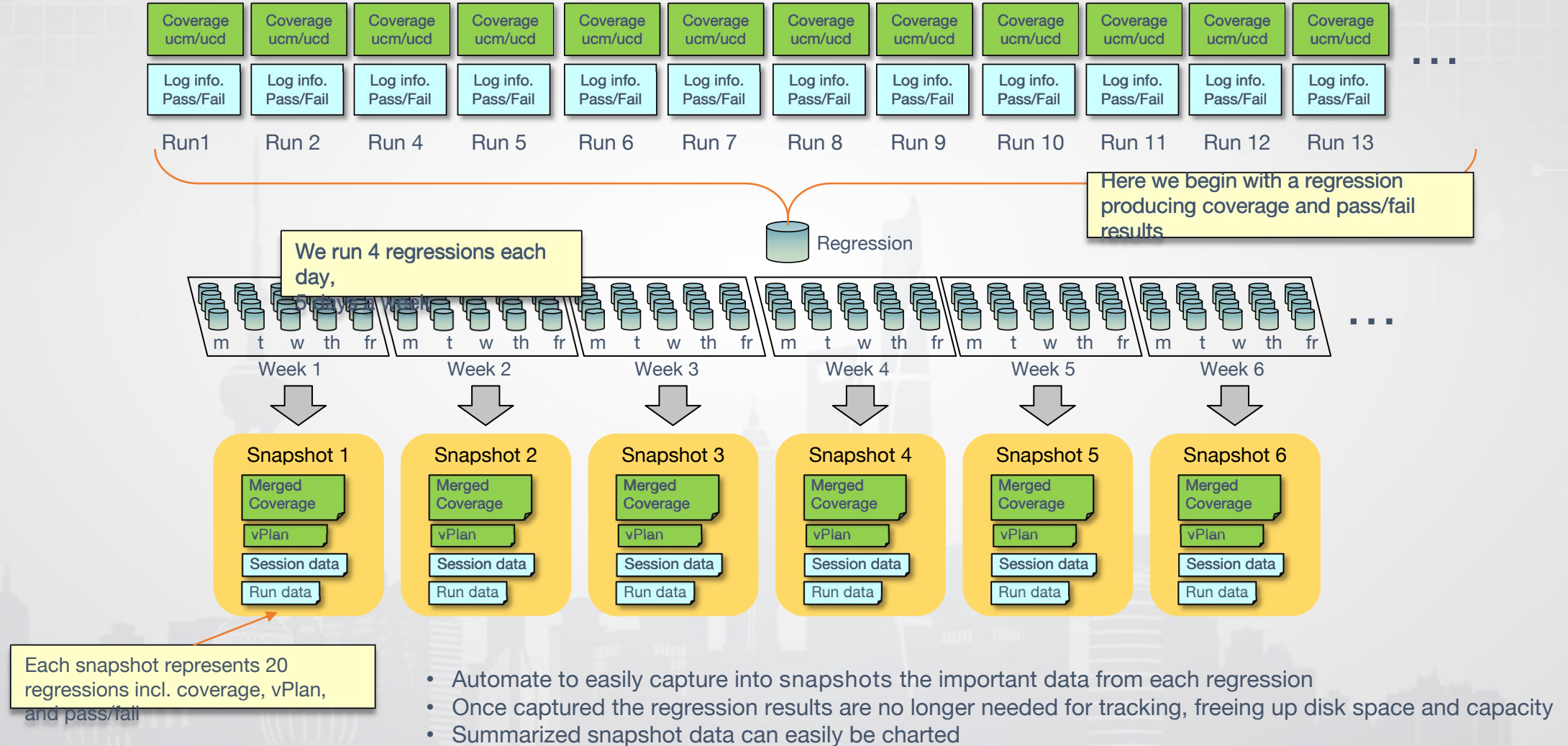


Simulation

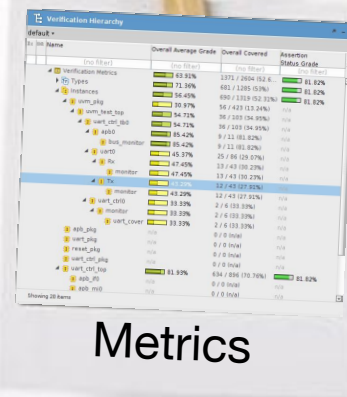
Formal

# Automation and Customization

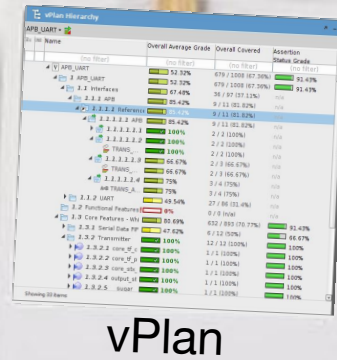
# Summarizing Regression Data – Tracking



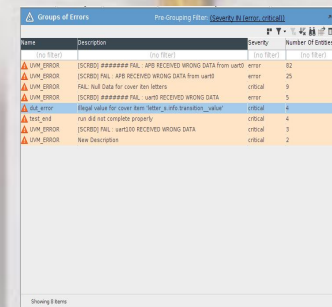
# What Data Is Interesting to Track?



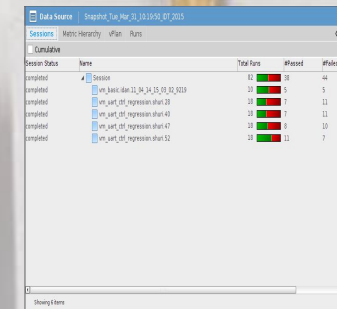
- Code coverage results for a particular module



- vPlan results for a particular perspective



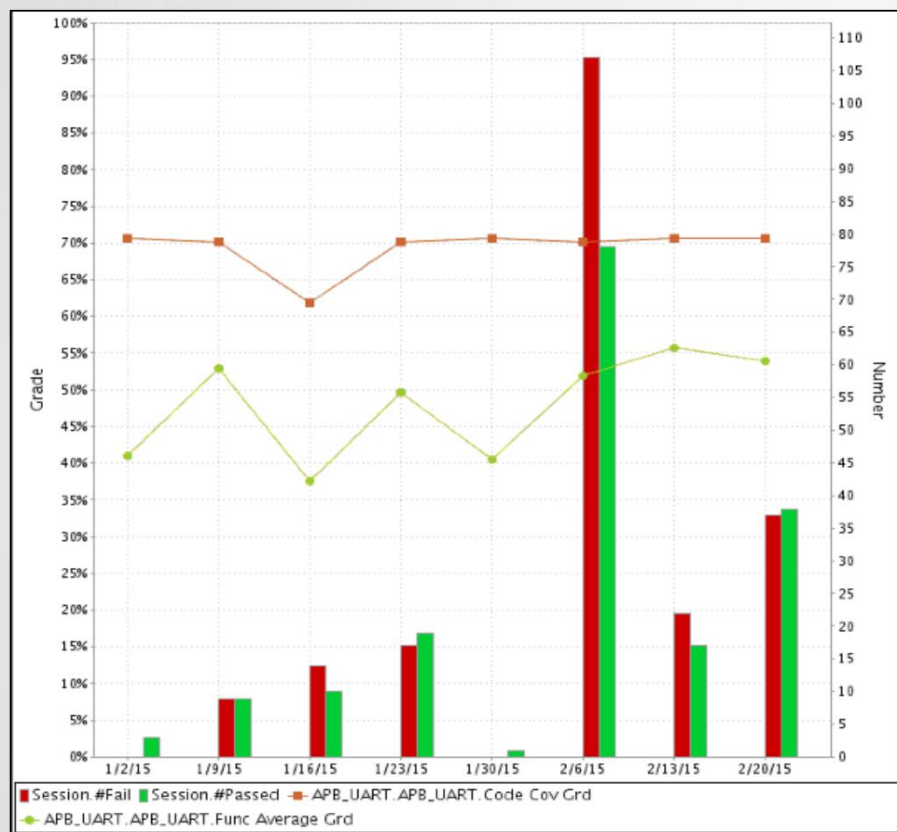
- Unique failures produced by each run for the entire project
- Who owns the most failures historically
- What are the most common tests in the regression



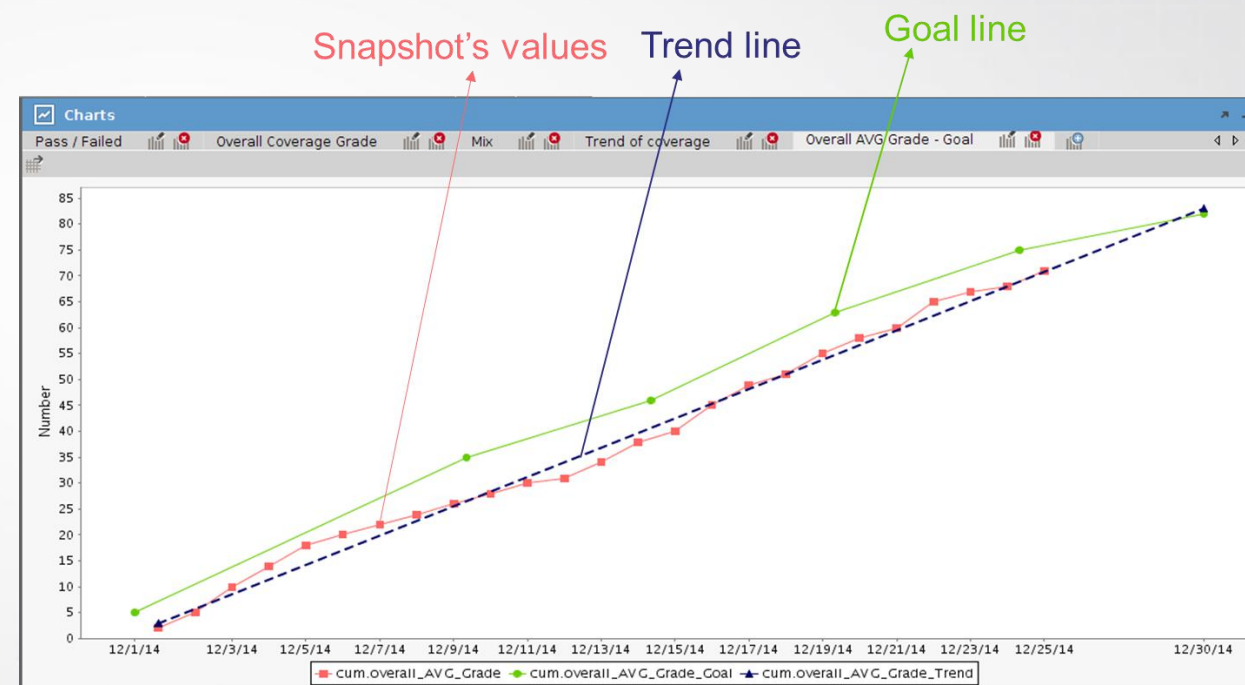
- Total number or runs, passed and failed

# Charting Tracked Data

## Quickly spot trends and anomalies



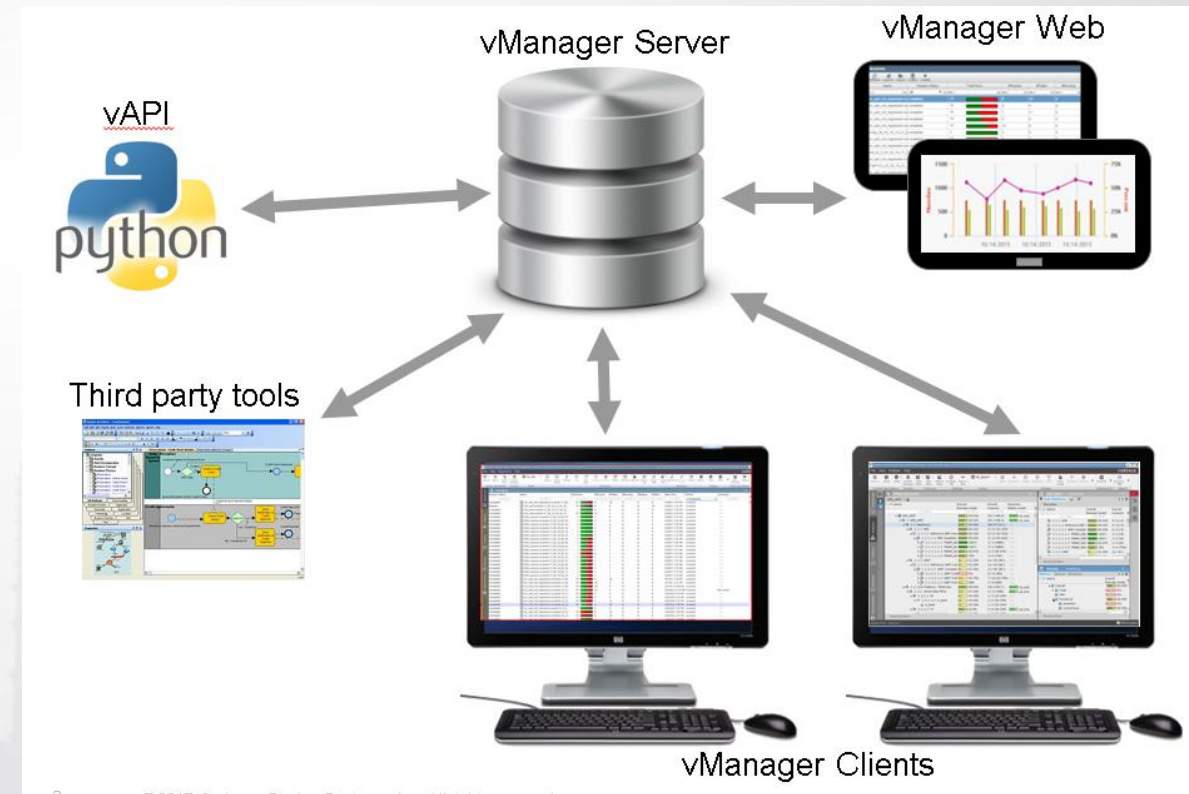
More Tests = More  
Coverage??



On Schedule??  
Converging??

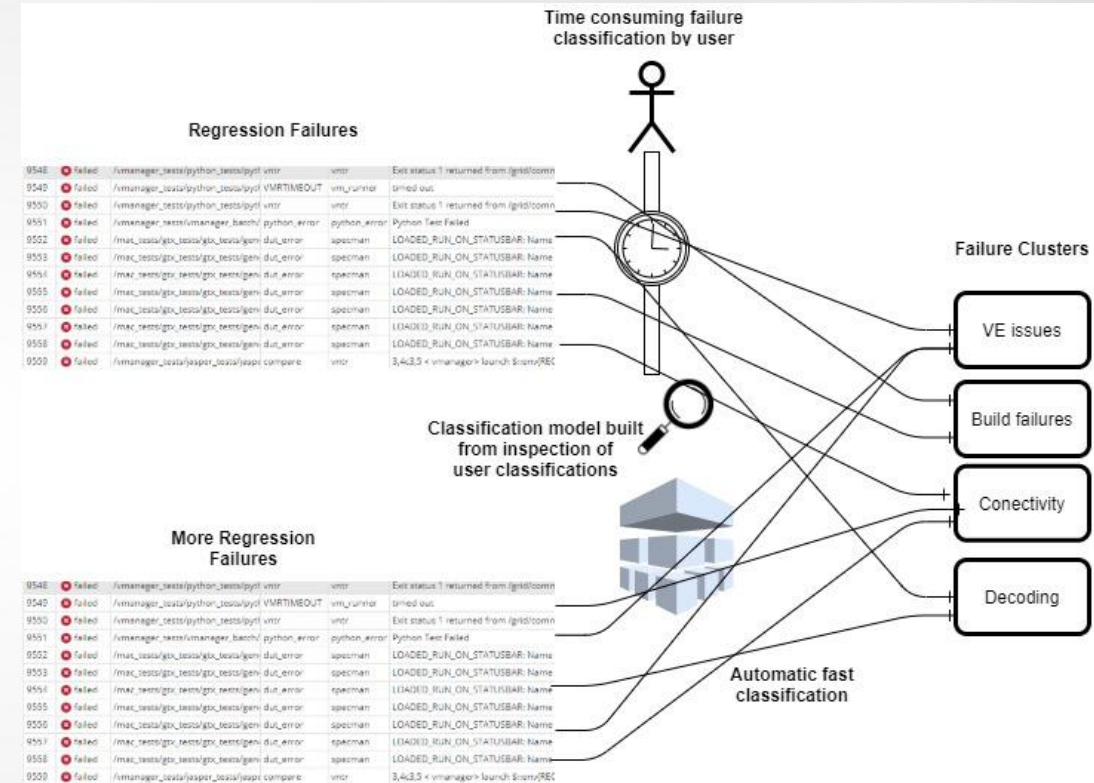
# Customization – vManager API: vAPI

- vAPI implements programmatic access to *Verification Management*
  - vManager™ database provides centralized data repository
  - Verification-specific abstraction of database content
- Query and mine verification specific data, with automation around tasks like coverage merge
  - “what is the toggle coverage on top.sig1 from all failing tests from last Tuesday”
- Execution of verification management actions on specific datasets
  - “rerun all tests that failed with parity error in the past 5 days”



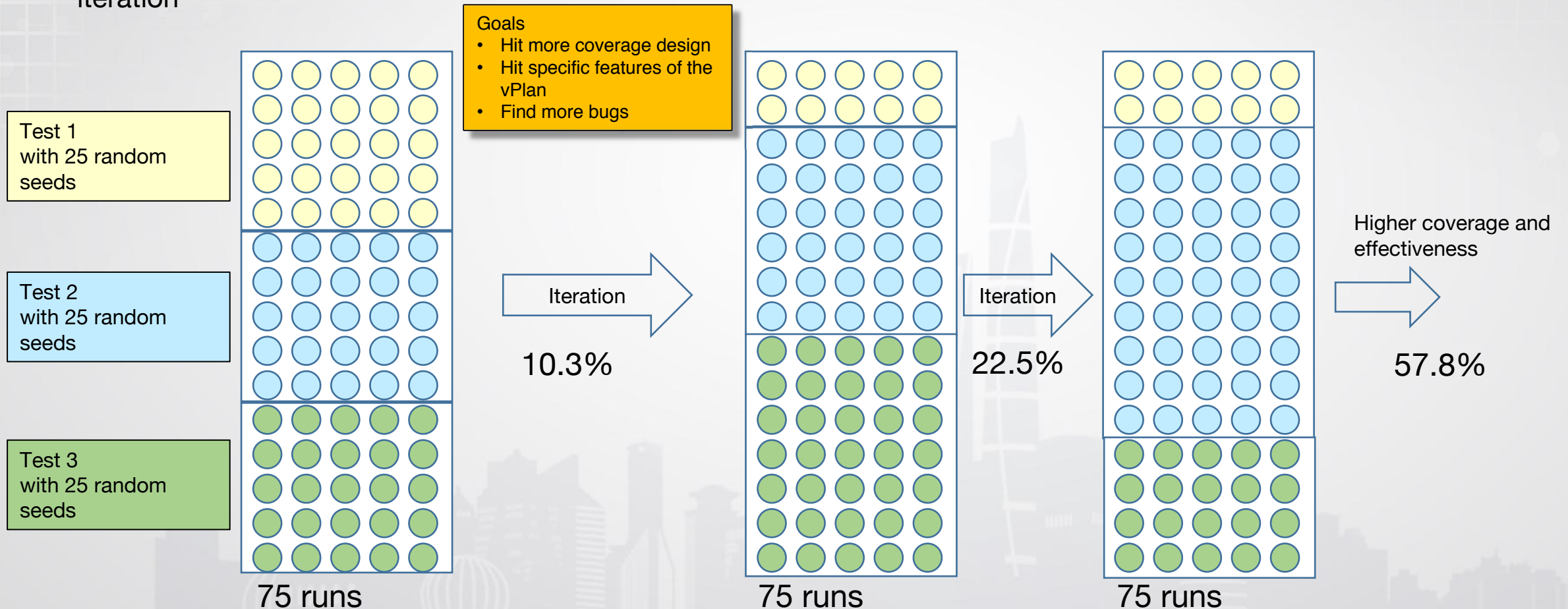
# Failure clustering using Machine Learning

- In today's daily regression failure analysis, the first analysis effort is to go over the failures and identify:
  1. Are they new?
  2. If they look alike, are they really the same as one found before.
  3. First Failure Analysis is not always sufficient:
    1. Error over-generic (Many bugs maps into same group)
    2. Error over-specific (Single bug maps into many groups)
- We are applying Machine Learning (ML) techniques by “learning” from the user manual assignments and “predict” future assignments automatically:
  - user assigns a failed run to a FC (Failure Cluster)
  - The system extract the properties of that run and consider them as characteristics of the FC
  - Whenever a run with “similar” characteristics would show up it could be candidate for assignment to the same FC.

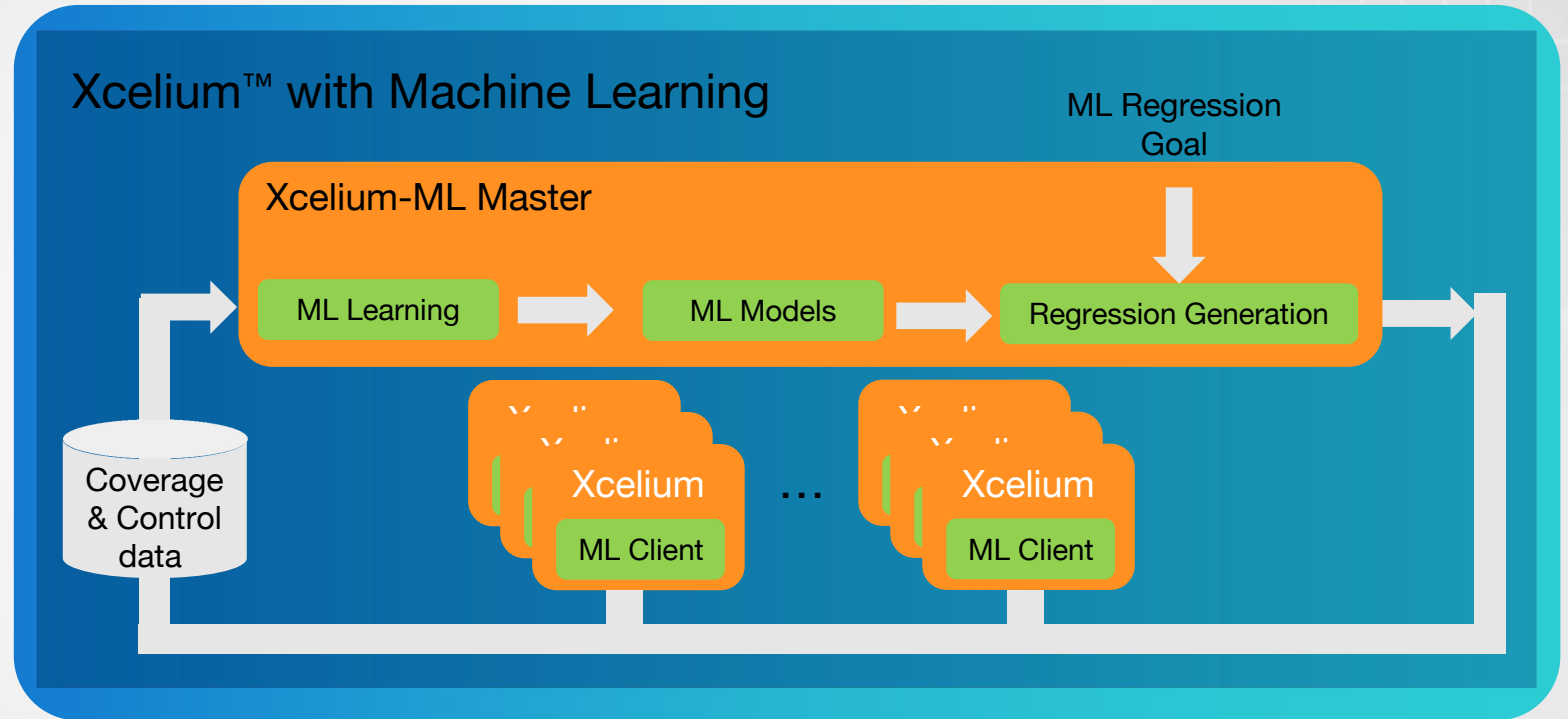
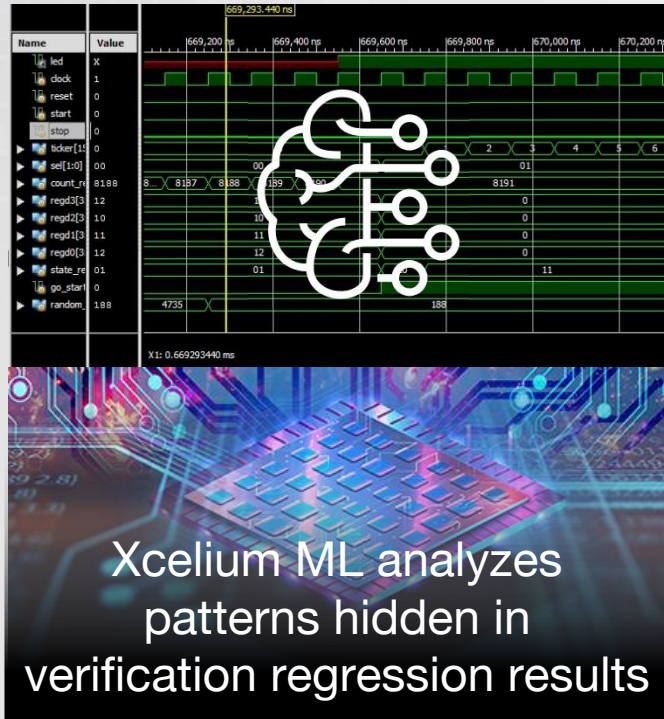


# Test Weight Optimization

- Identify and adjust the weighting of stimulus targeting a specific verification goal
- Each iteration results are evaluated against goals, effective tests are given more weight in the next iteration



# Xcelium ML User Flow



# Xcelium ML – User Feedback



I had yet to find \*any\* user comments about ML being applied to logic simulators for regressions nor anything simulation related.

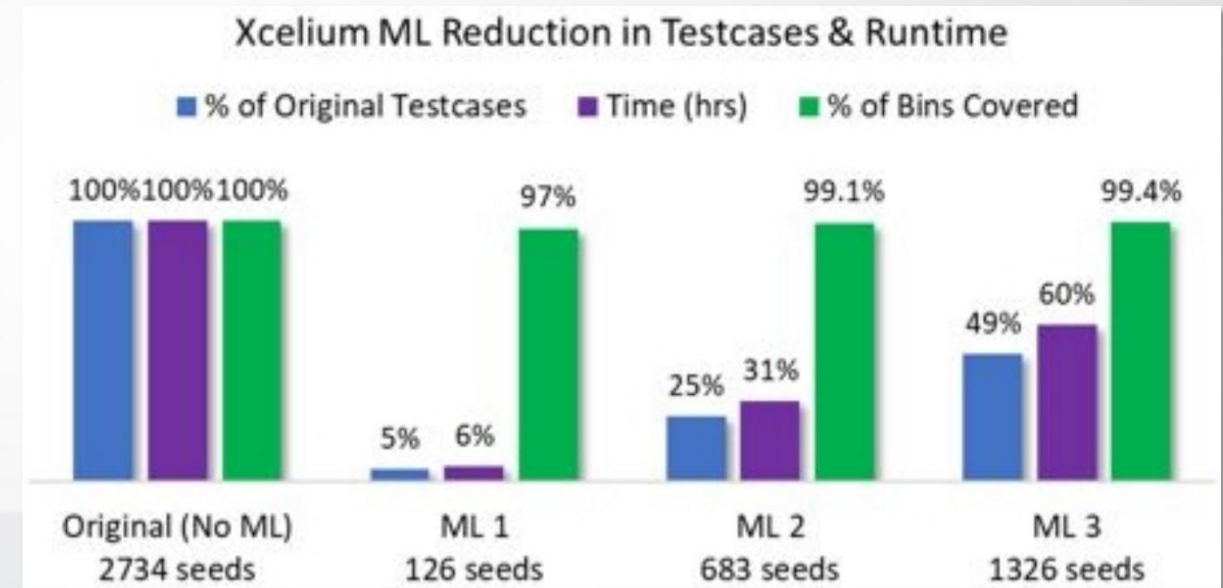
That is, until now.



In this year's report, two early ML users share that **Xcelium-ML got them a 2.5X to 3X speed-up in regression runtimes** -- with comparable coverage compared to their constrained random approach.

*"Xcelium-ML helped us generate a 3X smaller regression set while retaining 99+% coverage."*

*"Xcelium-ML improved our regression runtimes by 2.5X vs. Xcelium."*



Read the Full report: <http://www.deepchip.com/items/dac20-02b.html>

# Wrap Up

# Smarter Verification Management



- Better Productivity, Predictability, and Quality need more than ever.
- Planning, Collaboration and Centralization needs continue to expand.
- Multi-User, Multi-Engine, Multi-Region, Multi-Project are not optional.
- Thank you!

What the EDA users  
REALLY think  
**DEEPCHIP**

“vManager saves 1hr/day/engineer”  
Users Best of 2020

# Q&A



# cā dence<sup>®</sup>

© 2020 Cadence Design Systems, Inc. All rights reserved worldwide. Cadence, the Cadence logo, and the other Cadence marks found at <https://www.cadence.com/go/trademarks> are trademarks or registered trademarks of Cadence Design Systems, Inc. Accellera and SystemC are trademarks of Accellera Systems Initiative Inc. All Arm products are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All MIPI specifications are registered trademarks or service marks owned by MIPI Alliance. All PCI-SIG specifications are registered trademarks or trademarks of PCI-SIG. All other trademarks are the property of their respective owners.