

Boost Verification Efficiency with VC Execution Manager

Taruna Reddy, Principal Product Manager
Raghuram S Gururaja, Applications Engineering Manager
Synopsys



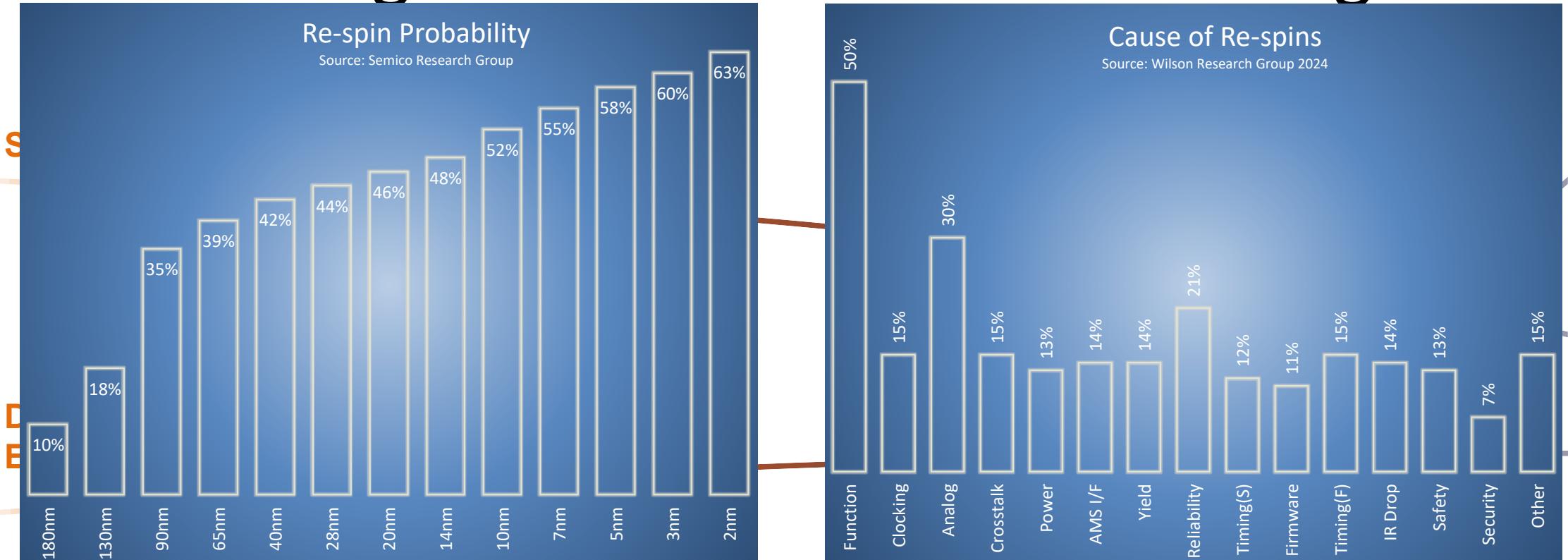
SYNOPSYS®



Agenda

- Introduction to VSO.ai (Verification Space Optimization)
- Verdi RDA (Regression Debug Automation)
- VC Execution Manager
- Benefits
- Demo
- Conclusion

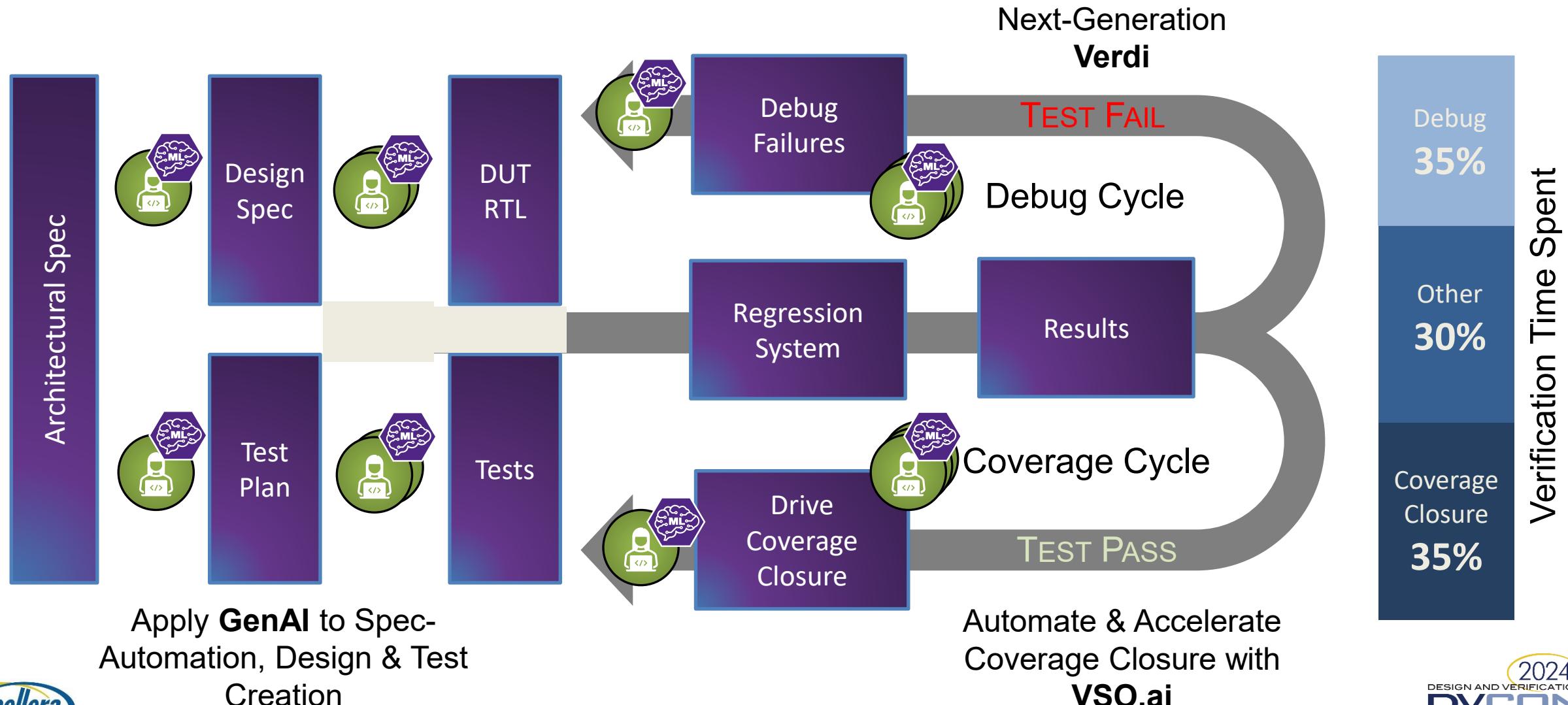
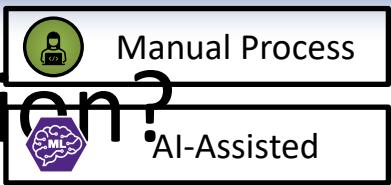
The Right First Time Silicon Challenge



Design size & Verification

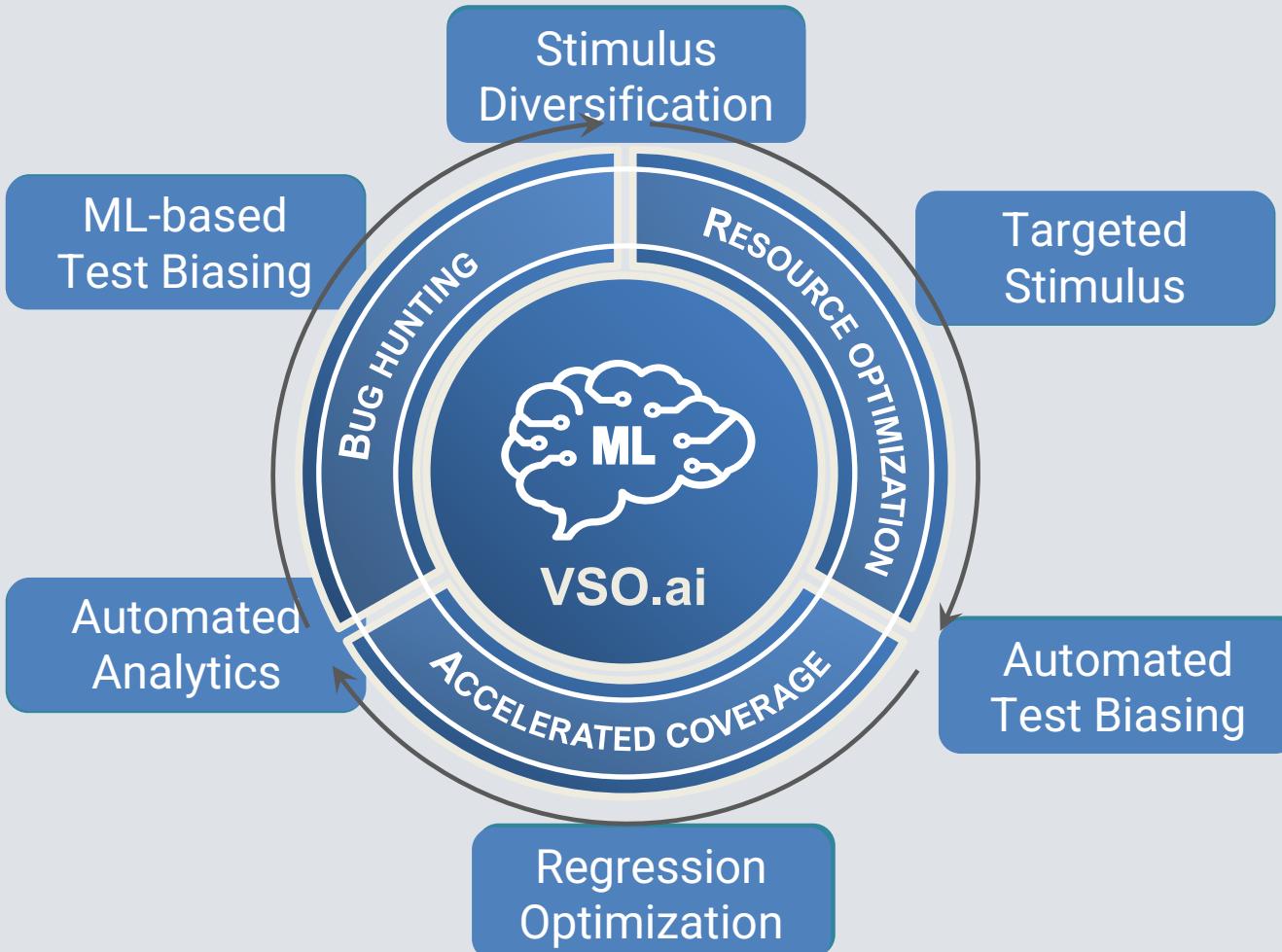
Find More Bugs with Limited Time & Resources

Where Are The Opportunities for Automation?



AI Driven Verification Space Optimization - VSO.ai

Faster, Higher Coverage Closure & Analytics



Stimulus Diversity: *Uncover bugs earlier while stressing the design*

Productivity Boost: *Advanced dashboards, metrics to analyze to converge on goals*

Improved HW Utilization: *Maximum coverage for the compute budget*

Higher Verification Efficiency: *High ROI tests to achieve coverage targets*

See What Customers Are Experiencing with VSO.ai

REGRESSION



Automotive
SoC



REDUCED
TAT BY
2x

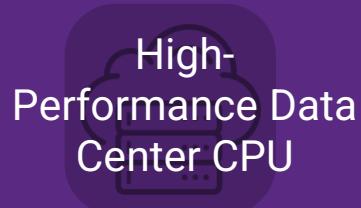
COVERAGE



IMPROVED BY
10%



HIGHER
QoR BY
50%



High-
Performance Data
Center CPU



REDUCED
TAT BY
16x



IMPROVED BY
5%



FASTER TTR IN
1st
ITERATION



High-
Performance
Wireless MCU



REDUCED
TAT BY
10x



IMPROVED BY
20%



2000
RARE COVERAGE
TARGETS IDENTIFIED



Mobile
SoC



REDUCED
TAT BY
8x



IMPROVED BY
3%



2600
ILLEGAL BINS
EXPOSED

VERDI RDA (REGRESSION DEBUG AUTOMATION)

Verdi Debug and Verification Management Platform

Widest-Adopted Debug Solution

Full-featured debug spans gates to software

Infused with AI-based root cause analysis

Integrated with Synopsys/third-party tools

Extendible with user-defined apps

Provides full verification management

Enables IDE extension for VS Code

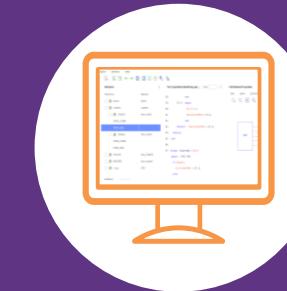
VERDI PLATFORM



Innovative
Debug



Verification
Management (VMS)



Integrated Development
Environment (IDE)

Unified Waveform and Coverage Data (FSDB, VDB)



Virtualization
Virtualizer
Platform Architect

Static/Formal
VC SpyGlass
VC Formal

Simulation
VCS

Emulation
ZeBu
ZeBu Empower

Prototyping
HAPS

VSO.ai

ICO

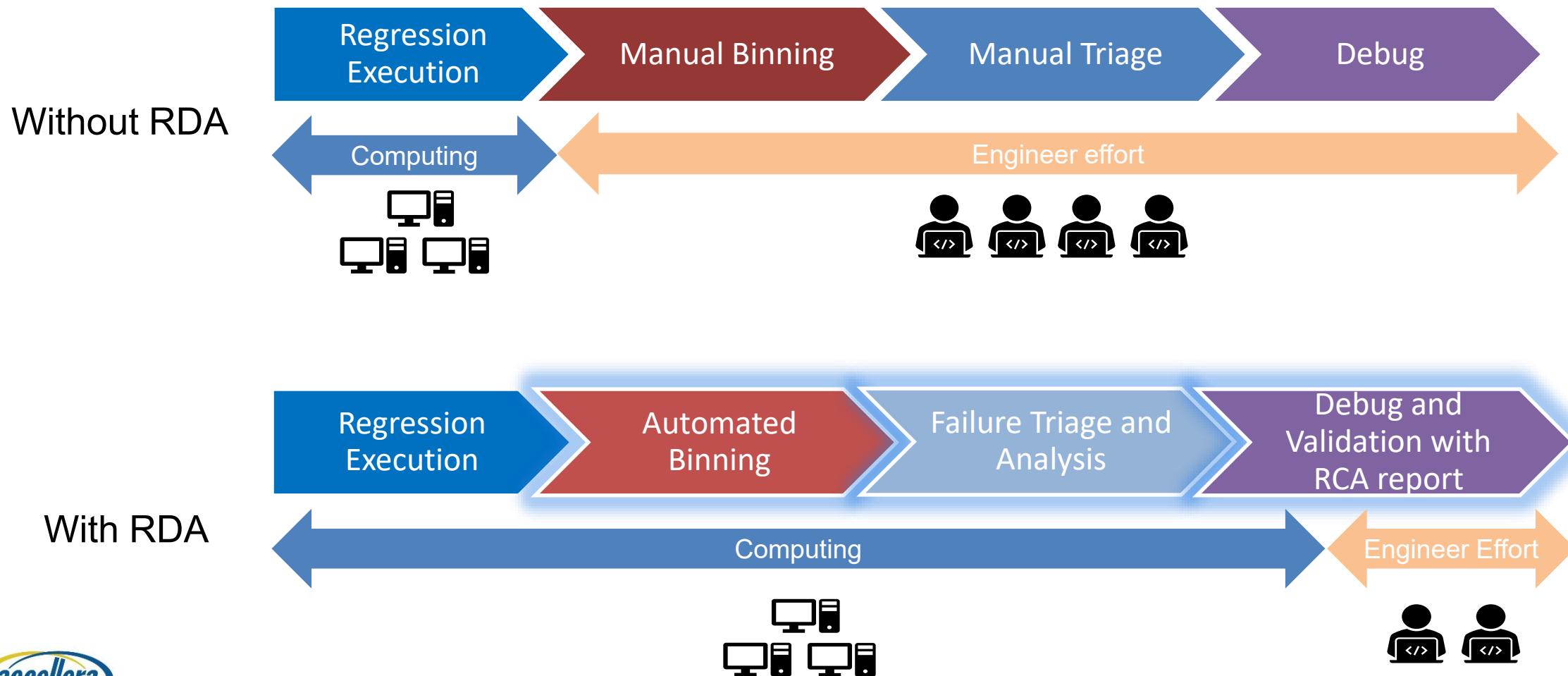
DPO

Verdi RDA

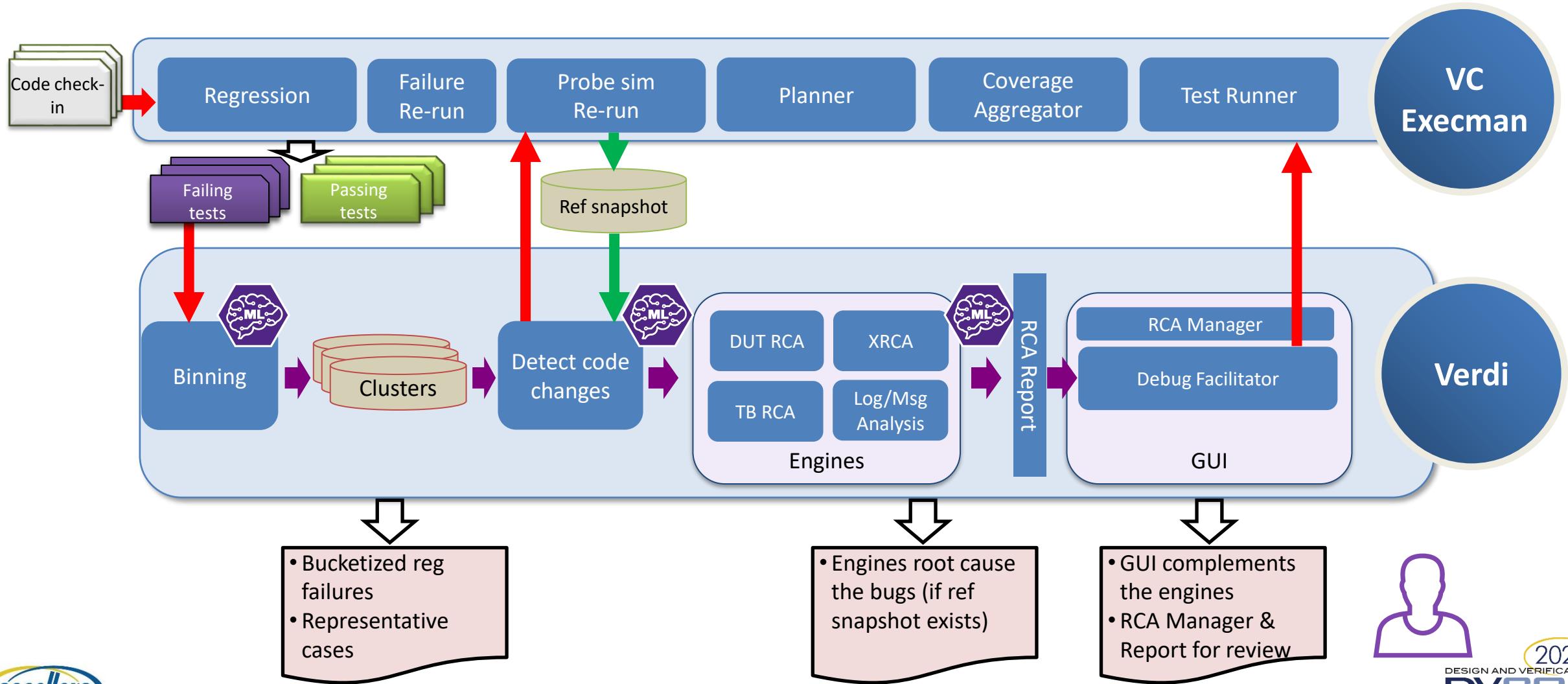


Regression Debug Automation (RDA) Motivation

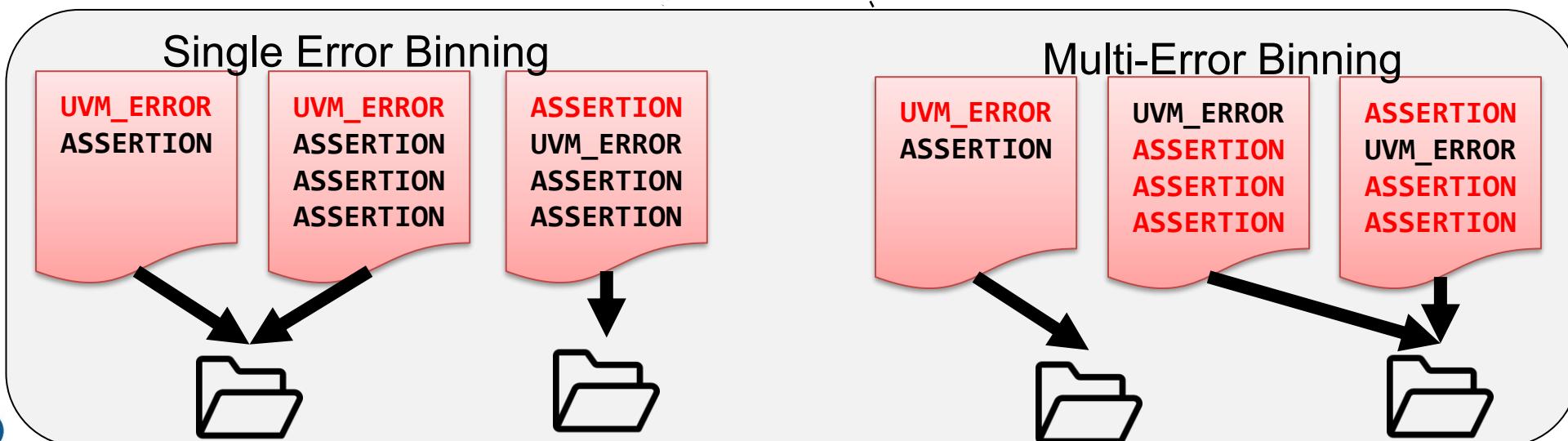
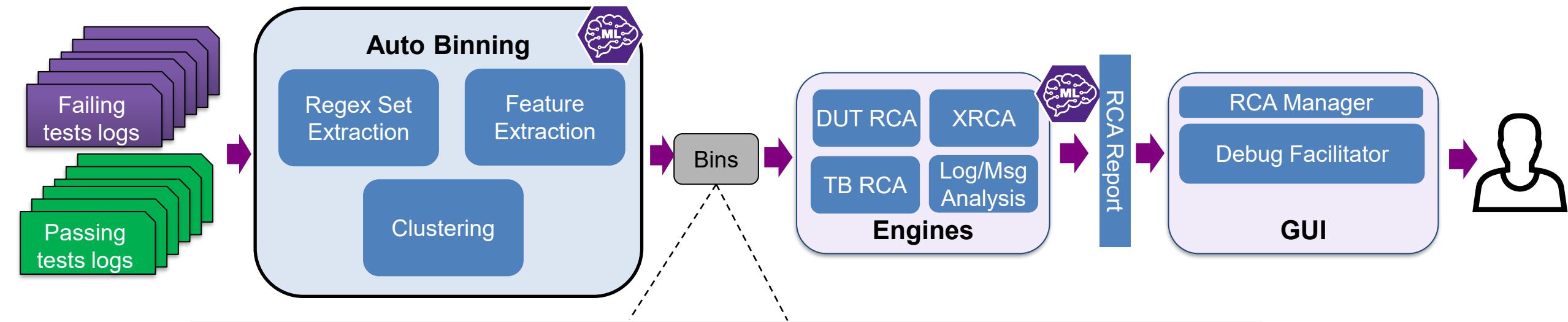
Reduce engineering effort/TAT with AI and advanced RCA technologies



ML-Based, Automated Regression Debug

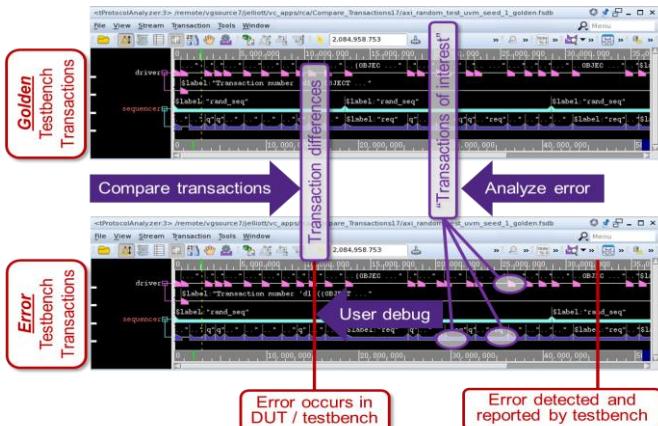


Verdi Regression Binning with ML



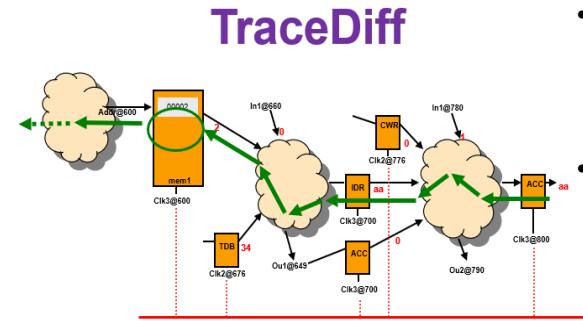
Verdi Root Cause Analysis (RCA) Engines

TBRCA



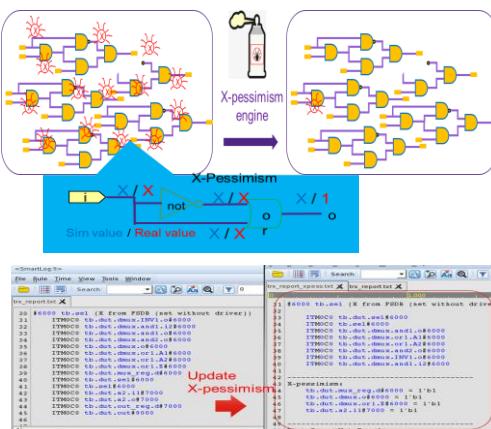
- Transaction Diff – Diff the transaction in the reference vs failing FSDB
 - Message Analysis – No ref FSDB required. Analyze info from the “error” message.
 - Report transaction of interest linked to the error

DUTRCA



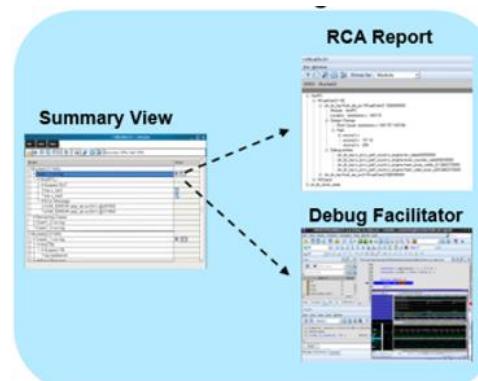
- Adopt roll back mechanism and TraceDiff technology to narrow down DUT problem
 - Temporal Flow View to analyze root cause path

XRCA / with X-Pessimism



- Scan X signals in FSDB and trace the root cause of X.
 - Handle large amount of X signals in batch mode
 - Formal engine to identify X pessimism to remove the noise

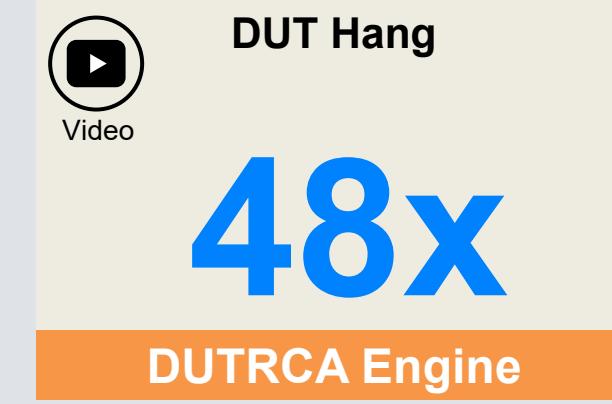
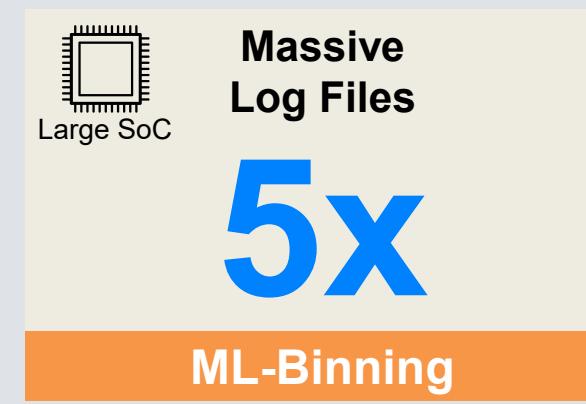
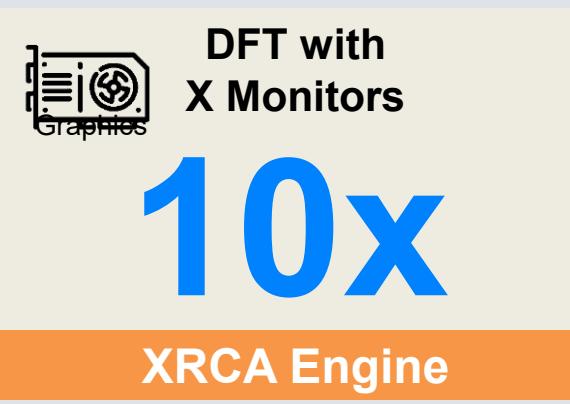
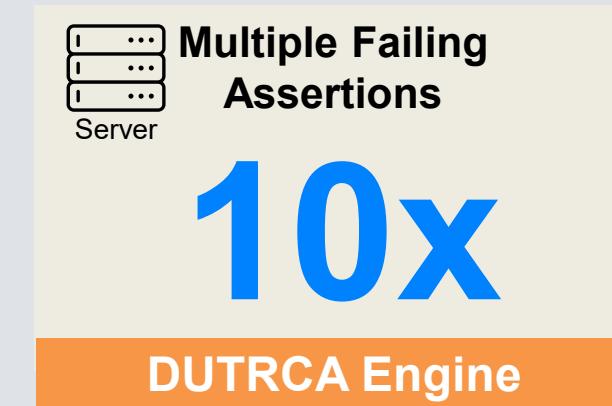
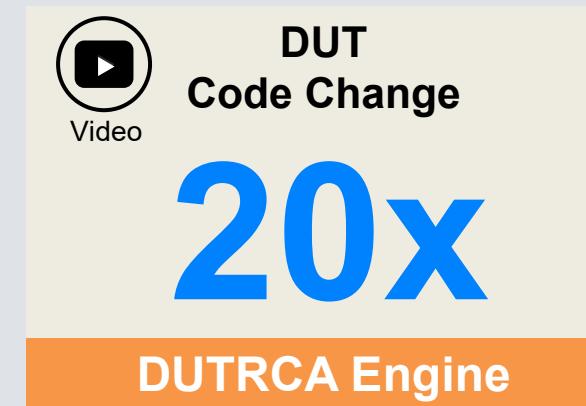
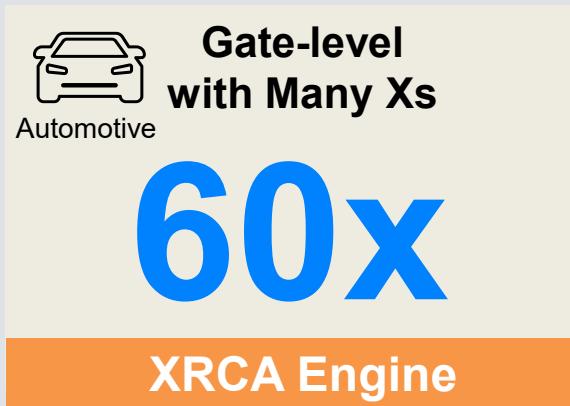
Debug Facilitator



- Generates debug data nightly for each bin.
 - Facilitates user to debug with “Protocol Analyzer” and “TB Reverse Debug”
 - Reduces about 30 ~ 40% debug effort for TB debug

Verdi Next-Gen: Accelerate Debug Automation

Customer Examples



VC EXECUTION MANAGER

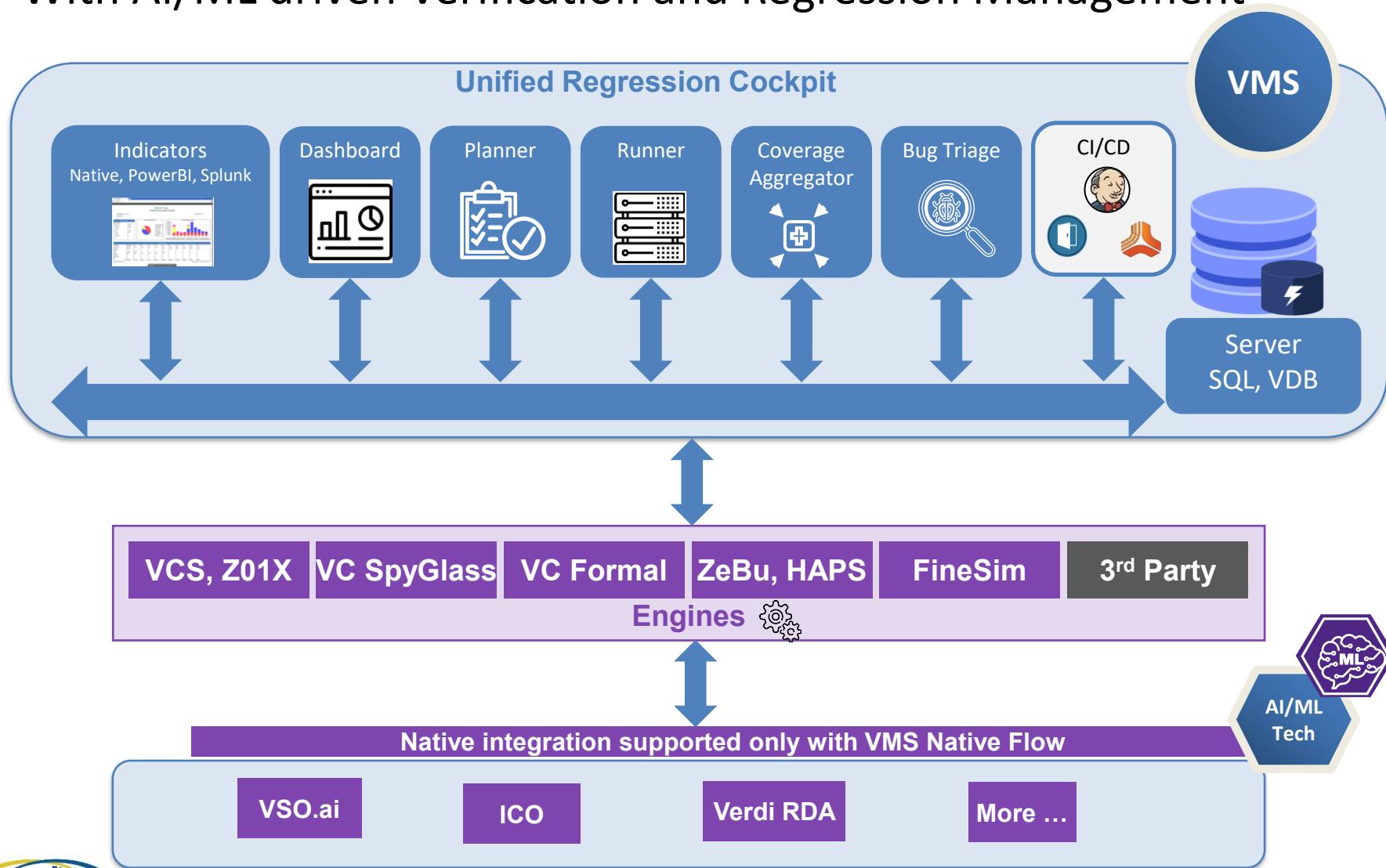
Synopsys
Confidential
Information

2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
INDIA



VC Execution Manager

With AI/ML driven Verification and Regression Management



- Cockpit GUI based on Verdi
- Native indicators with open database support for PowerBI & Splunk.
- Comprehensive dashboards
- Scalable multi-user Verification/Coverage Planner
- Unified and extensible Runner
- 24x7 Coverage Aggregation
- Native failure binning, triage and debug assistance
- Unified AI-enabled technologies
- Enables CI/CD methodology support for verification
- Cloud support
- Supports industry standard APIs – Rest, CLI, Python etc.

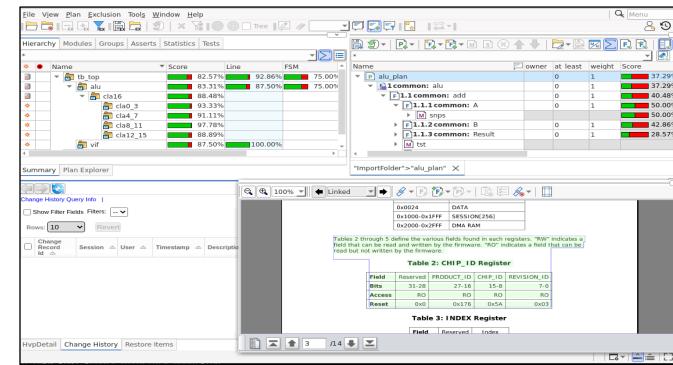
VC Execution Manager – Key Modules

Dashboard



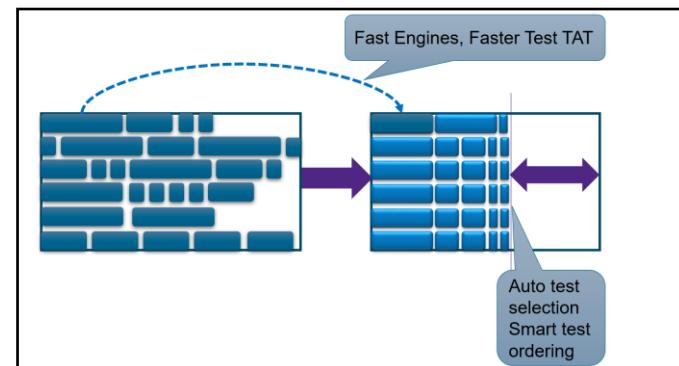
- Test planning, execution & debug, coverage merge and annotation
- Enables verification data-over-time to be mined for analytics

Planner



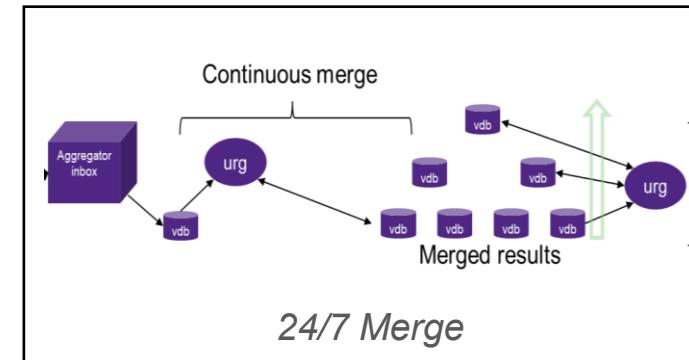
- Multi-user test scheduling/planning
- Supports change history and restore
- API for automated report generation and updates

Runner



- Runs and optimize regressions
- Order tests to eliminate long tail

Coverage Aggregator



- Continuously merges incoming coverage
- Integrated tagged VDB from ad hoc regression runs
- Can generate moving window merge VDB

Dashboard Users View – Usability

The screenshot displays three windows of the Verdi tool:

- Session Attributes:** Shows a list of sessions with columns for Name, Status, Start Time, RTIME(s), Hostname, Log File, Comment, Cmdline, Error, Test Time, Tags, Action, Test Trend, and History.
- Protocol of read:** A waveform diagram illustrating the read protocol with signals for clk, reset, csn_N, ramDat, and ramA_ddr.
- CovDetail:** A detailed view of coverage analysis, showing cover group items, score, instances, and U+C metrics for various conditions like operand_A, operand_B, result, and range comparisons.

VC
Execution
Manager

API

Engineer View

The screenshot shows two main panels of the Synopsys Execution Manager:

- Trend Chart:** A line graph titled "trend2" showing performance metrics over time for different components: Wishbone::Assert, Wishbone::Cord, Wishbone::Group, and Wishbone::Line. The Y-axis represents INDEX (Percentage) from 60% to 100%.
- Session Management:** A table listing sessions with columns for User, #Sessions, Latest Session, Tot, P, F, W, A, U, Summary, Created, Test Trend, and Plan Trend. Sessions include bart, etrend, alu_uvm, abc, saves_vdb, merge_ex, sriwatsa, filespew, two_builds, and simple.

Management View

- Verification plans created/viewed by multiple users
- Organizes content by projects and projects hierarchically
- Scalable for large designs and users and cloud compatible
- Smart editor with rich text support, inline images, tables; bulk edits; API

Planner

DB based Verdi Hierarchical Verification Planner (HVP)

The screenshot displays the Synopsys Planner application interface. The top navigation bar includes File, View, Plan, Exclusion, Tools, Window, and Help. Below the navigation bar are several toolbars with icons for file operations, search, and other functions.

The main window is divided into several panes:

- Hierarchy:** A tree view showing a hierarchy of verification plans. Nodes include tb_top, alu, cla16, cla0_3, cla4_7, cla8_11, cla12_15, and vif. Each node has columns for Name, Score (e.g., 82.57%), Line (e.g., 92.86%), and FSM (e.g., 75.00%).
- Plan Explorer:** A detailed view of the alu_plan. It shows a breakdown of the plan into common, add, A, B, Result, and tst components, each with its own score and weight.
- Change History:** A table showing a list of changes with columns for Record Id, Session, User, Timestamp, and Description. A dropdown menu allows filtering by filter fields.
- Registers:** A detailed view of the CHIP_ID Register. It shows memory ranges: 0x0024 (DATA), 0x1000-0x1FFF (SESSION[256]), and 0x2000-0x2FFF (DMA RAM). A note states: "Tables 2 through 5 define the various fields found in each registers. 'RW' indicates a field that can be read and written by the firmware. 'RO' indicates a field that can be read but not written by the firmware." Below this is Table 2: CHIP_ID Register, which defines fields for Bits (31-28), Access (RW), and Reset (0x0). There is also a reference to Table 3: INDEX Register.

- **Database backed application** enables real time collaboration
- **Typed features** ensures that all plans satisfy project template requirements
- **Rich text fields** to describe detailed descriptions
- **Project-wide queries and bulk updates**
- **Change history and restore**
- **API for automated report generation and updates**
- **Verdi UI (Linux) and Web UI** (all platforms)

Planner - Details

The screenshot displays the Synopsys Planner application interface, which includes several windows and toolbars:

- Plan Explorer**: Shows a hierarchical tree view of project components under "vita_0119_planner". A blue callout points to this window.
- Hvp**: Shows a list of feature types and their details. A blue callout points to this window.
- HvpDetail**: Shows a detailed table of attributes for a specific feature type, with a red box highlighting the table area. A blue callout points to this window.
- Richtext Viewer (on odclega)**: Shows a rich text editor with a red box highlighting the toolbar. A blue callout points to this window.
- Project Detail**: Shows a table of identifier, attribute limit, and child limit. A red box highlights the table area. A blue callout points to this window.
- Plan Detail**: Shows a table of feature types, attributes, metrics, and spec files. A blue callout points to this window.
- Change History**: Shows a table of changes with columns for Feature, Old Section, and New Section. A blue callout points to this window.
- Exclusion Manager**, **Plan Search**, **Project Detail**, **Message**, **Change History**, and **Restore Items**: These are tabs at the bottom of the interface.

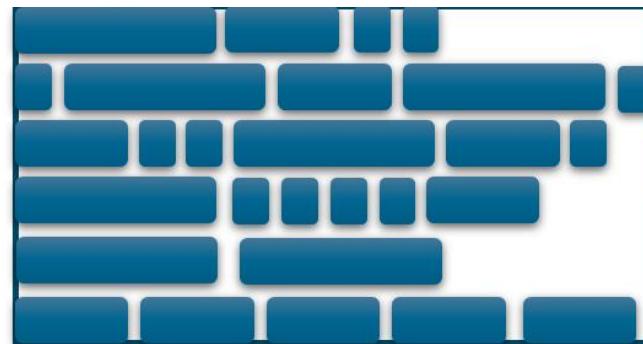
Synopsys Confidential Information

Optimize Regressions

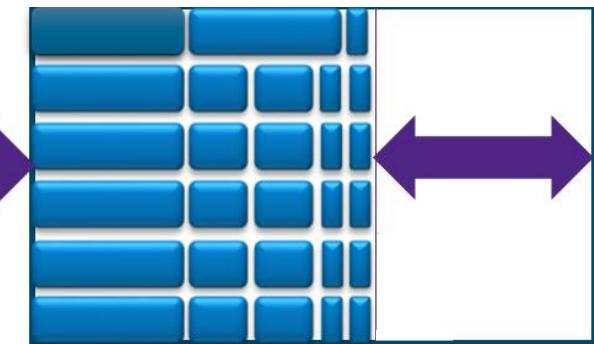
Runner

- Reduce regression TAT and resources
- Schedules tests based on history to eliminate long tail
- Improves compute resources utilization, reducing costs
- Native cloud support

Scheduled Tests (Before)



Scheduled Tests (Optimized)

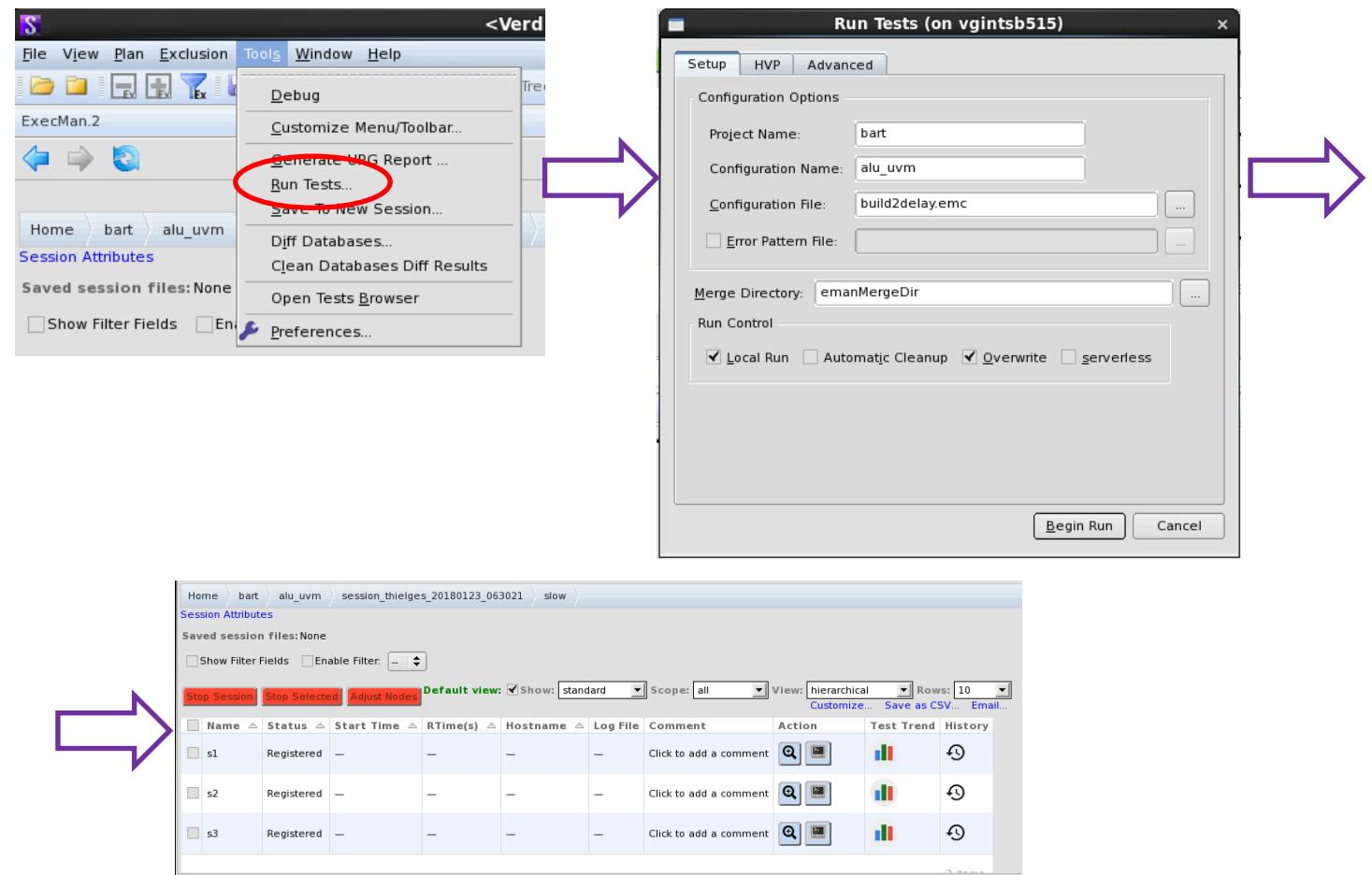


Considers: coverage, engine speed, smart test ordering

Runner - Launching Regressions

Methods

- Interactive
 - Easiest way to launch regressions
 - Includes live controls to kill, rerun, and adjust sessions
- Batch
 - Useful as part of complex scripts or launched as cron jobs



- Coverage from additional verification tools can be incorporated
- Annotate VC Formal results in the coverage database
- Display VC Formal assertion status in the coverage report
 - Verdi coverage and URG
- Measure VC Formal assertion status in HVP

Coverage - Beyond Simulation

VC Formal example

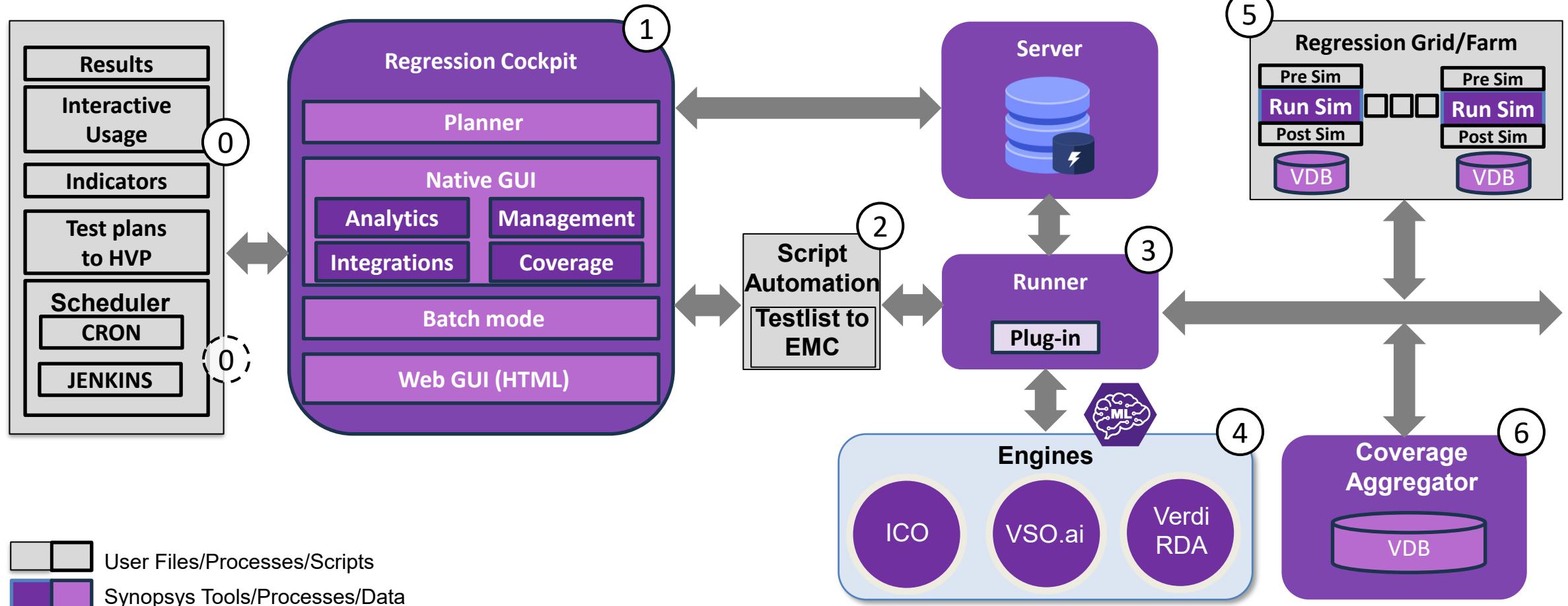


AI/ML INTEGRATION AND BENEFITS

Synopsys
Confidential
Information

2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
INDIA

Integrated View: VC ExecMan with VSO.ai & Verdi RDA



Unified Verdi VMS with AI/ML Technologies

Optimized verification with reduced resources, risks and maintenance

Ease of Use and Data Analytics via Integration

- Faster rollout to project teams (save up to 12 months)
- Seamless integration with native solutions like VSO.ai
- Unified cockpit solution for regression results, data analytics, optimization outcomes and health of ML model

Reduced Risk, Simpler Maintenance

- Reduces incremental support overhead from weeks to days
- One DB/server (vs. multiple machines/DB/servers for hybrid systems with glue logic)
- Up to 2X reduced hardware costs

Regular AI/ML Rollouts and Updates

- 2-10X higher QoR with integrated ML and verification technologies when having full data access/control
- Immediate deployments of updates of VSO.ai, RDA, DPO, ICO and VC Formal rollouts (vs. months of integration effort)

DEMO

Conclusion

One-stop solution to ease the complexity of diverse verification tasks

Accelerates the deployment of advanced AI/ML verification and debug technologies

Scales to accommodate multiple users and manage regressions seamlessly

Results made simple with a user-friendly, graphical view of results