WEBINAR

Managing System-on-Chip (SoC) Complexity: Strategies for Scalable Design

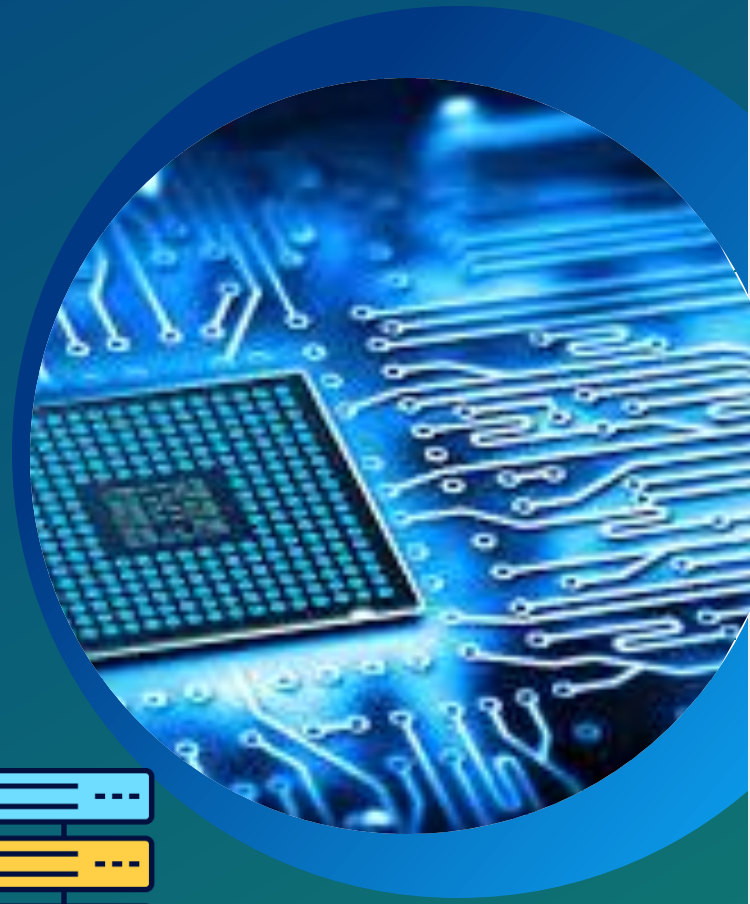
Steve Rush

Principal Solutions Consultant

Jama Software

Agenda

- The Challenge
- Strategies for scaling IP and SoC complexity
- Demonstration
- Q&A



What are they?

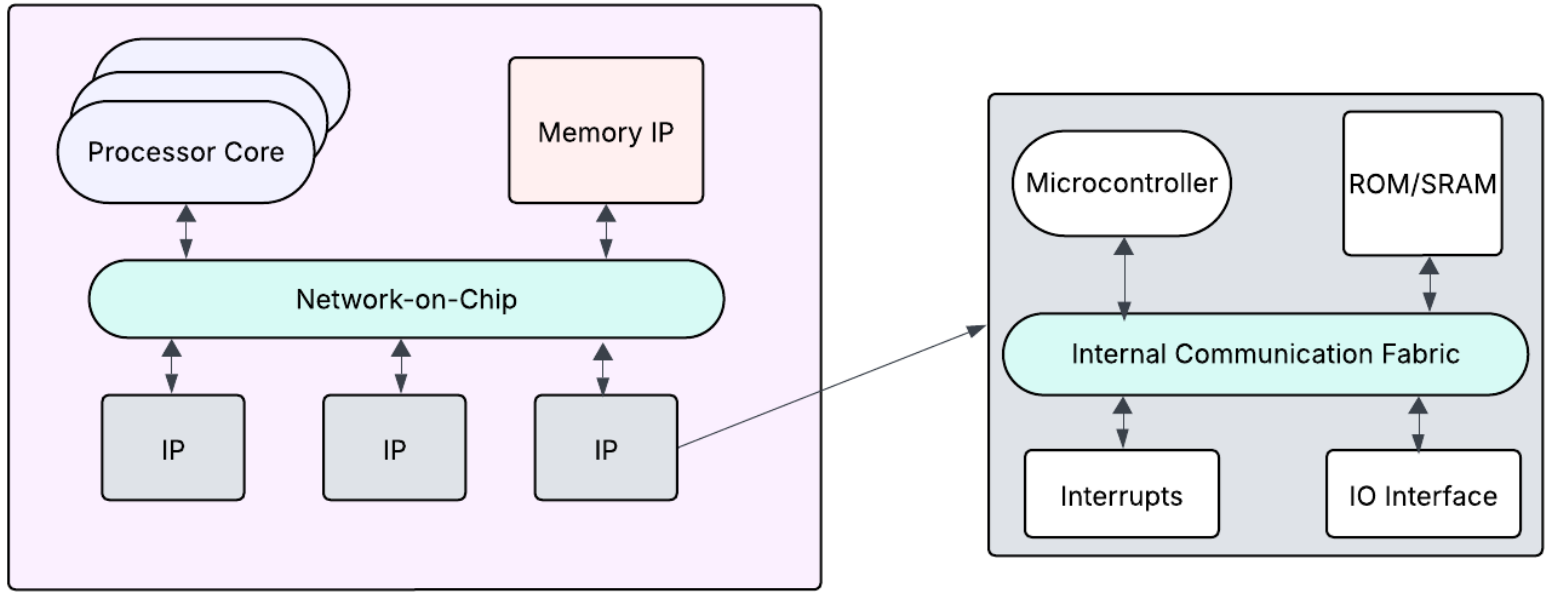
IP & SOC

IP (Intellectual Property):

- IP cores are foundational elements in the design of systems-on-chip.*

SoC (Systems-on-Chip):

- A system on a chip is an integrated circuit that compresses all of a system's required components onto one piece of silicon.+

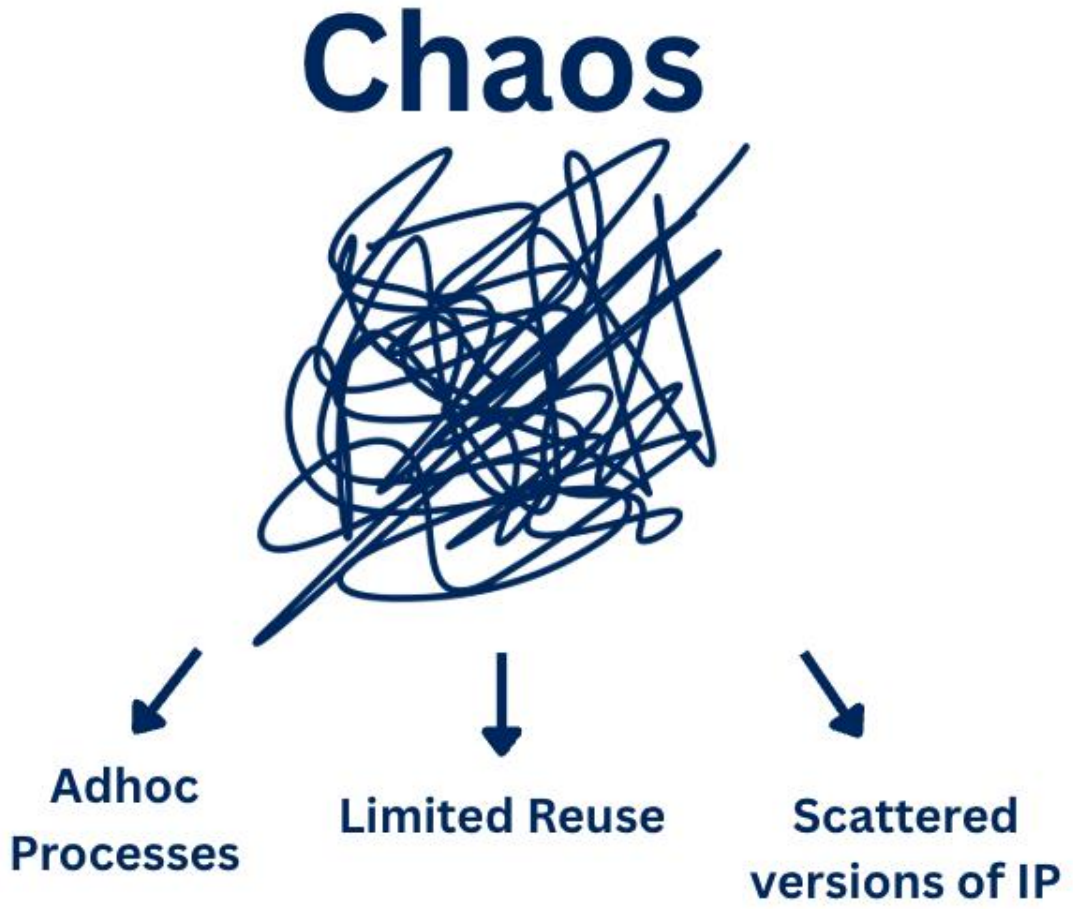


Source: Medium

Sources: *AnySilicon, +ANSYS

The Challenge

- Each IP project has its own process, strongly leveraging institutional knowledge.
- IP core requirements maintained in separate, inconsistently organized projects
  - Sometimes maintained in disparate systems, with no unifying data model
- Low or no reusability
  - Engineering data (requirements included) must be created from scratch, leveraging institutional knowledge



What I hear among customers

Current Challenges

- Need to scale SoC management in Jama
- Different versions of IP scattered across multiple projects/systems
- Maintaining licensing for 3rd party IP
- Future projects coming down the pike which will require reuse of IP

Goals

- Scale SoC management
- Leverage libraries and variant approaches to compile IPs for reuse
- Immediate version clarity– both for licensing and defect/change management

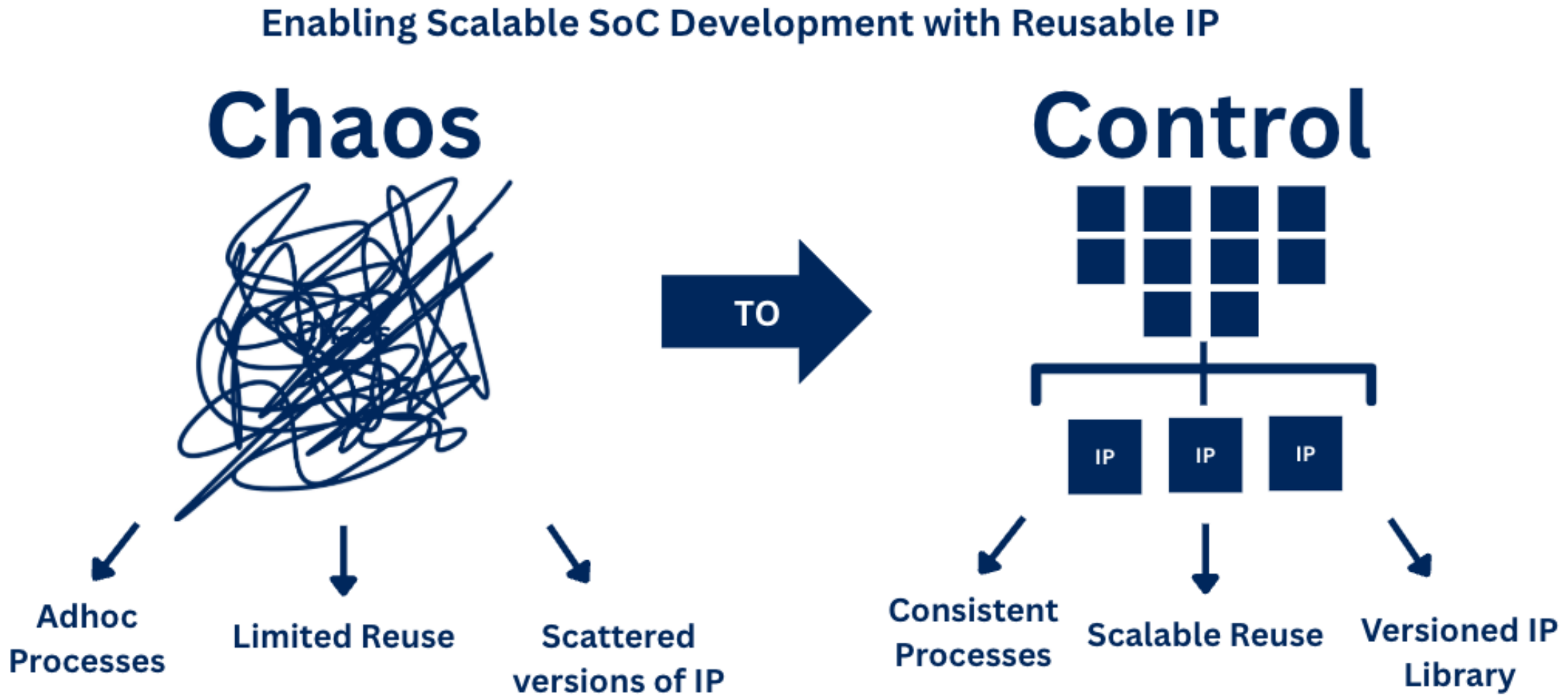
Impact

- Managing IP across different projects leads to disorganization
- Traceability risks
- Future projects will take longer to develop due to rework, recompiling of existing IPs

Business Outcomes

- Streamline SoC management leveraging existing IP
- High quality source control using library approach
- Maintain licensing and regulatory compliance
- Increase time to value, manage change, increase product confidence

From Chaos to Control



JAMA CONNECT™ PRODUCT DEVELOPMENT PLATFORM

SOLUTION COMPONENTS

REQUIREMENTS MANAGEMENT

- Test & Quality Management
- Risk & Hazard Analysis

KEY FEATURES

- Live Traceability
- Review Center
- Real-time Collaboration
- Reuse & Baseline Management
- Workflow & Configuration Management
- Import/Export Wizard
- Integrations via REST API
- Visibility & Compliance Reporting
- Customer Support & Community
- Flexible Deployment Model (Cloud & Self- Hosted)

INTEGRATIONS | PROFESSIONAL SERVICES | ADDITIONAL FEATURES

Strategies for Managing IP and SOC Complexity

Strategies for Managing IP and SOC complexity

DATA MODELS AND TEMPLATES

- Unified yet configurable data model
- Structured starting point with flexible configurability, standardized permission scheme

REUSE (VIA LIBRARIES)

- Version Managed IP blocks, mix and match reusability
- Recreate requirement, test, design content and relationships in a matter of minutes
- Managed changes and source control

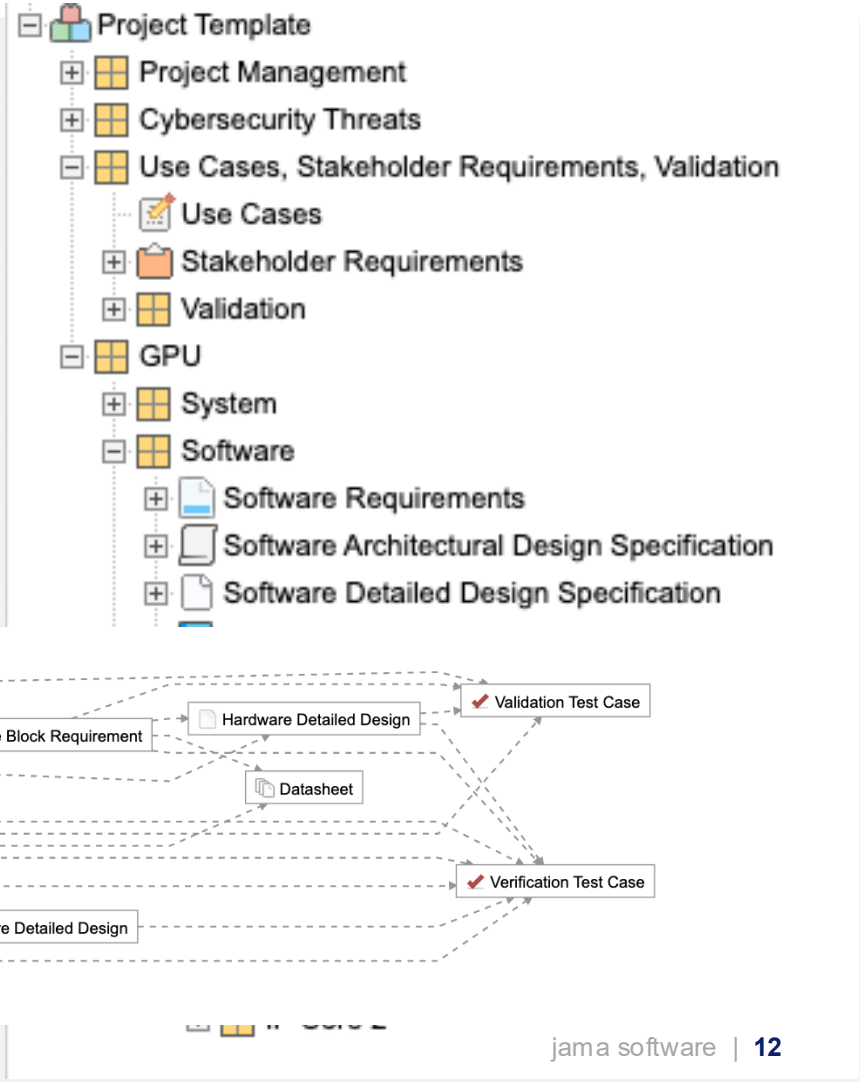
LEVERAGING INTEGRATIONS

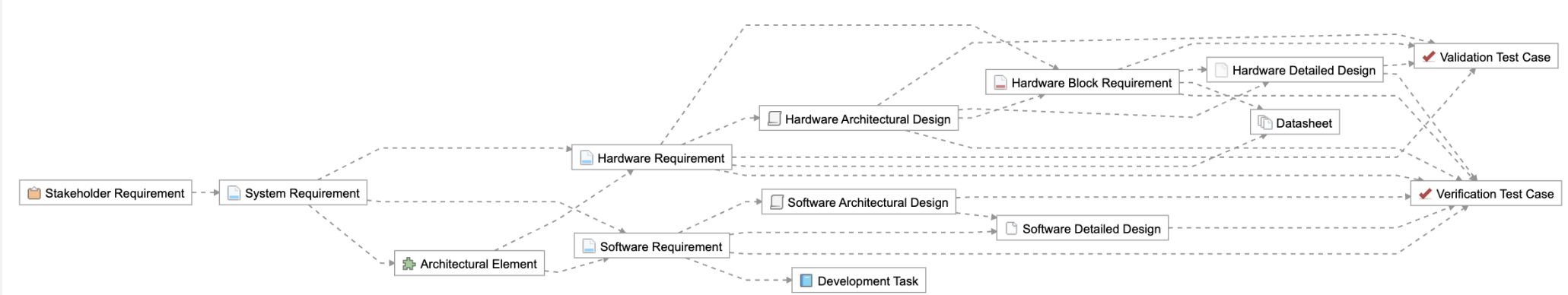
- Leverage best-of-breed tools and integrations to achieve Live Traceability
- Pre-built integrations and REST API

Strategy 1 – Data Models and Templates

Project Templates and Data Models

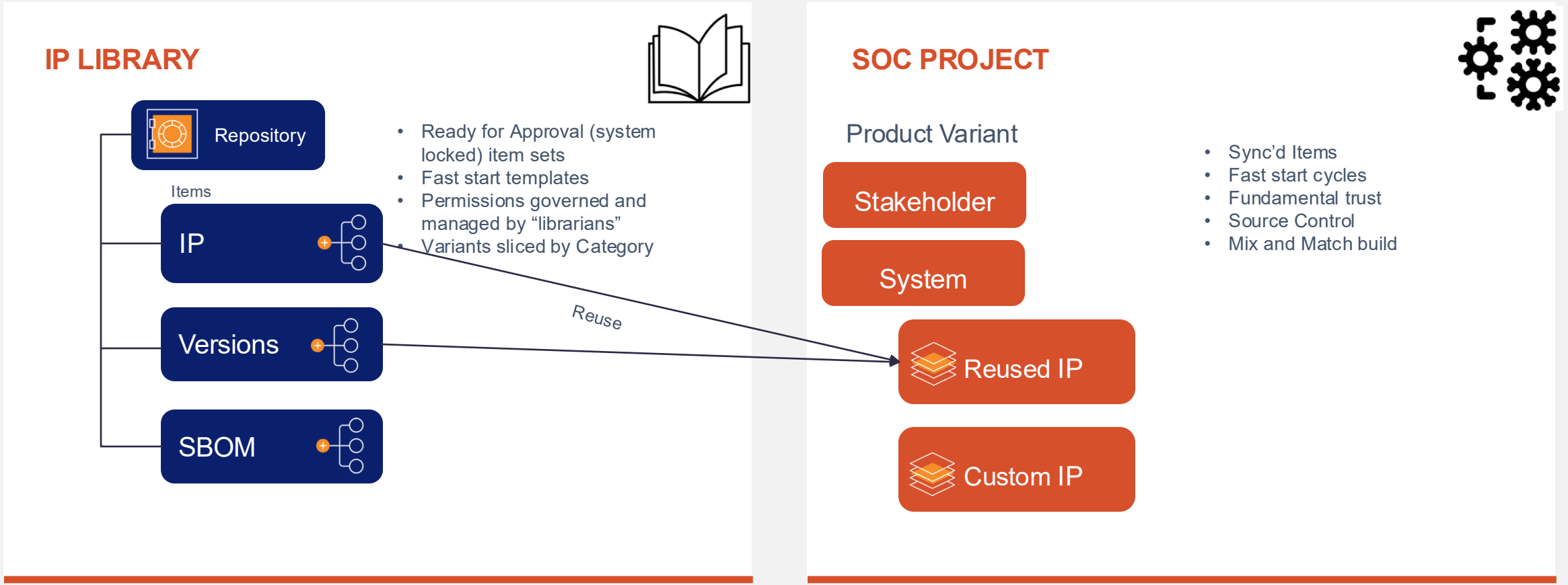
- Hierarchical Organization structure for your content
- Location based and visual, drag and drop capabilities
- Permission scheme allows content access
- Traceability Information Model defines allowed relationships





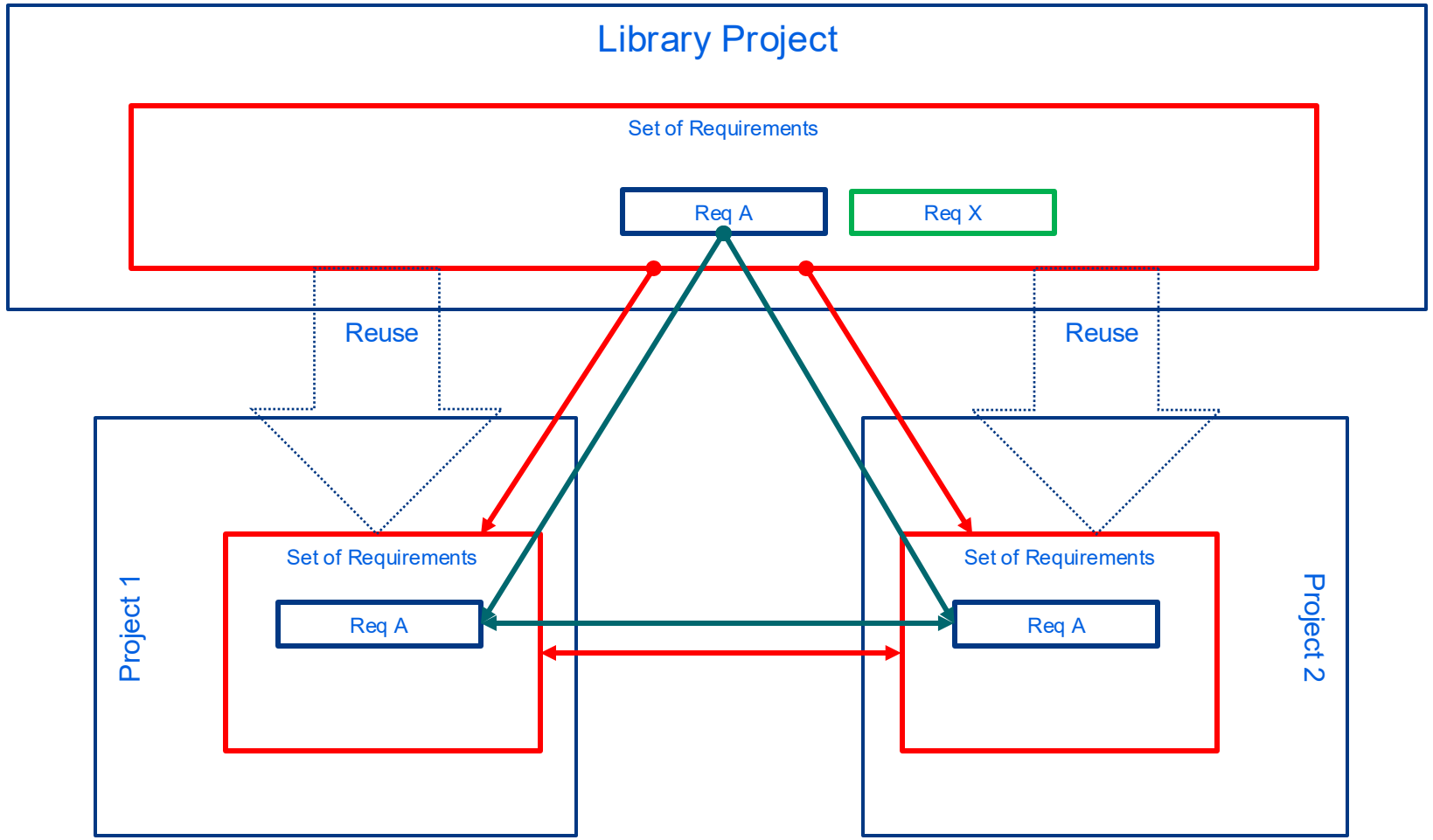
Strategy 2 – Reuse

IP Centric Reuse



Source Control

REUSE & SYNCHRONIZATION



Approach:

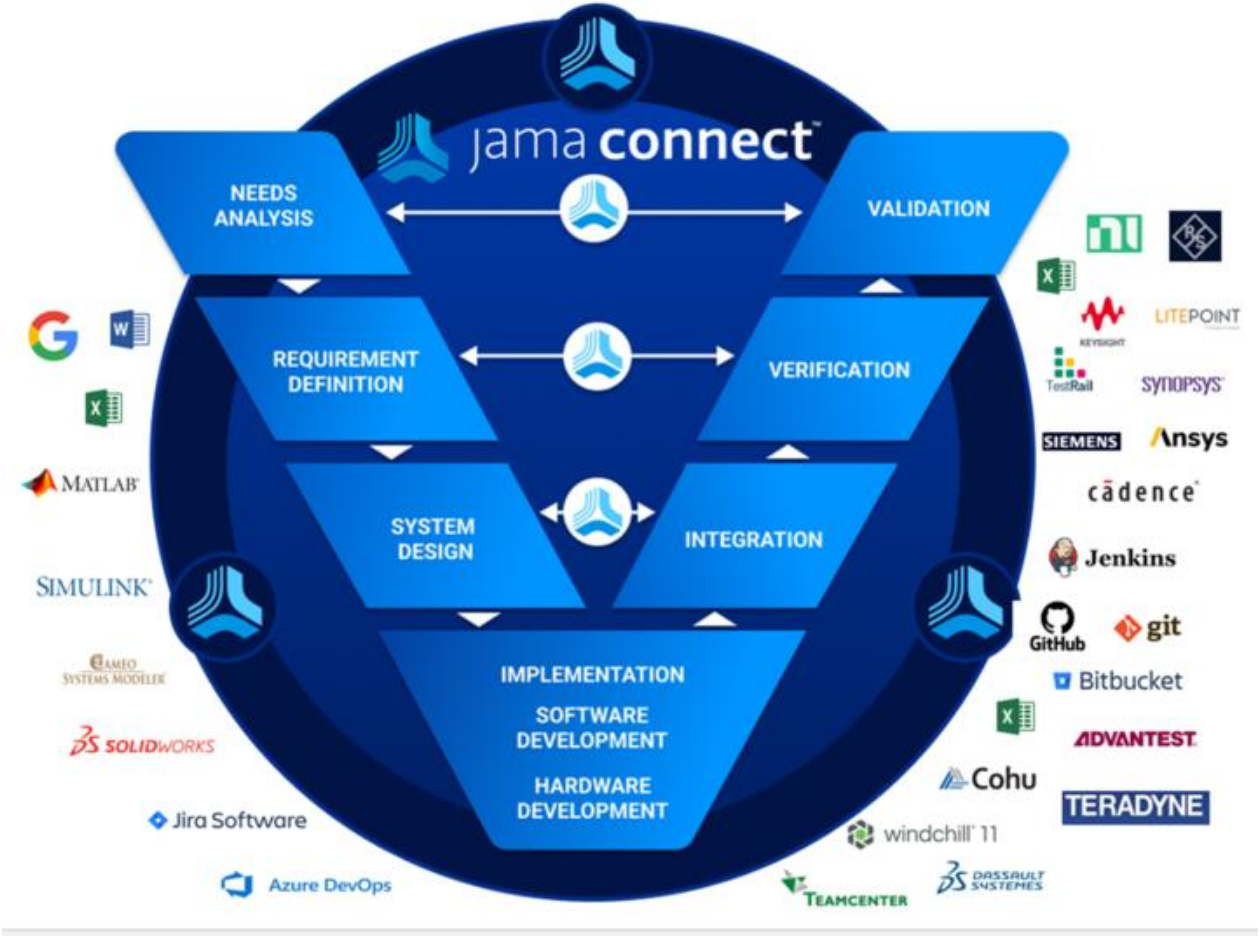
- Library project houses collection of requirements maintained by “curator” (permission-based)
- Projects reuse from the library with synchronization enabled to monitor for differences.
- Curator of library can pull diffs back into library if desired
- Project members can pull updates from library into projects.

Strategy 3 – Integrations

Best-of-Breed Toolchain

LEVERAGING INTEGRATIONS

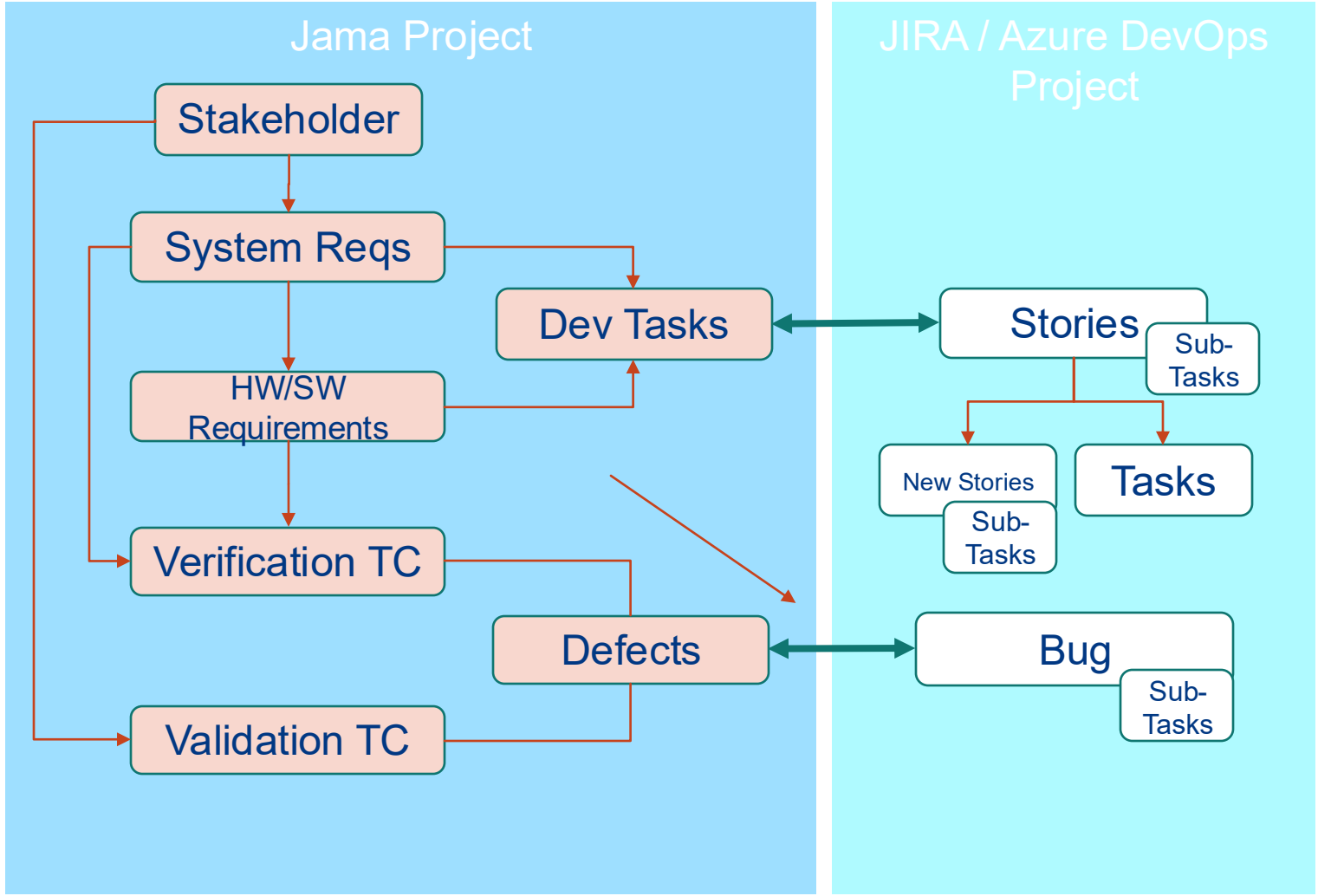
- Achieve Live Traceability™ by easily navigating upstream and downstream relationships to understand impact of change across development lifecycle and best-of-breed toolchain.
- Integrate, don't duplicate
- Partner integrations, integration hubs and one-off integrations using the REST API are all possible



Integration to Task Management:

RECOMMENDED PATTERN

- Stakeholder Requirements, System requirements, HW/SW Requirements and test cases controlled in Jama Connect.
- No task management in Jama Connect, instead requirements are linked to an implementation story used to provide context.
- In Jira, teams break up the work into issues that are linked to the implementation story.
- Defects discovered in testing are tracked for resolution and re-test



Q&A

Thank You