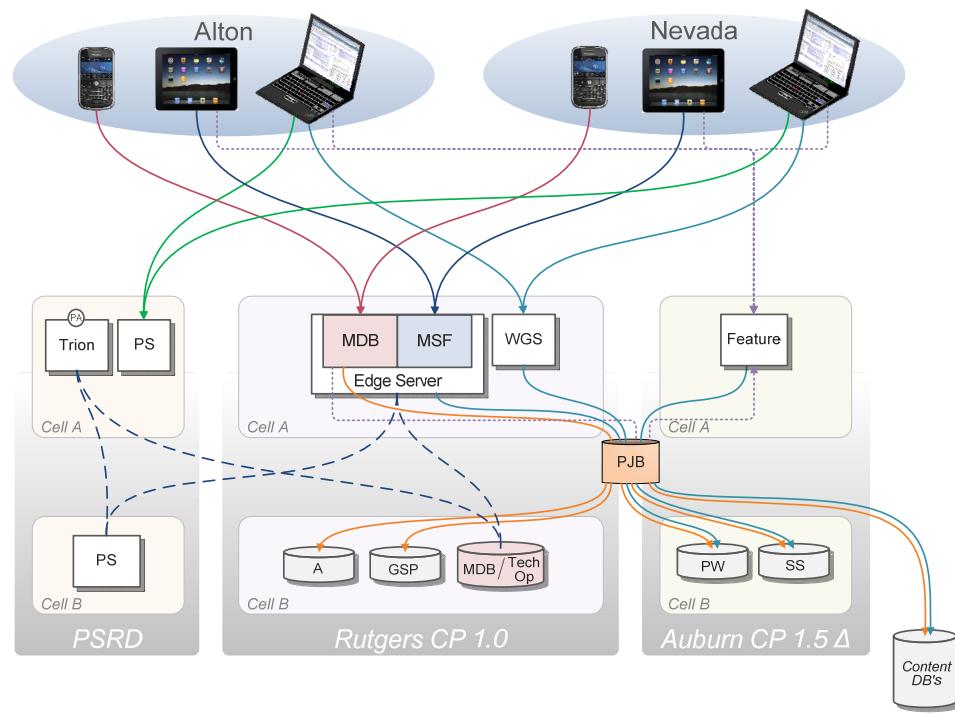


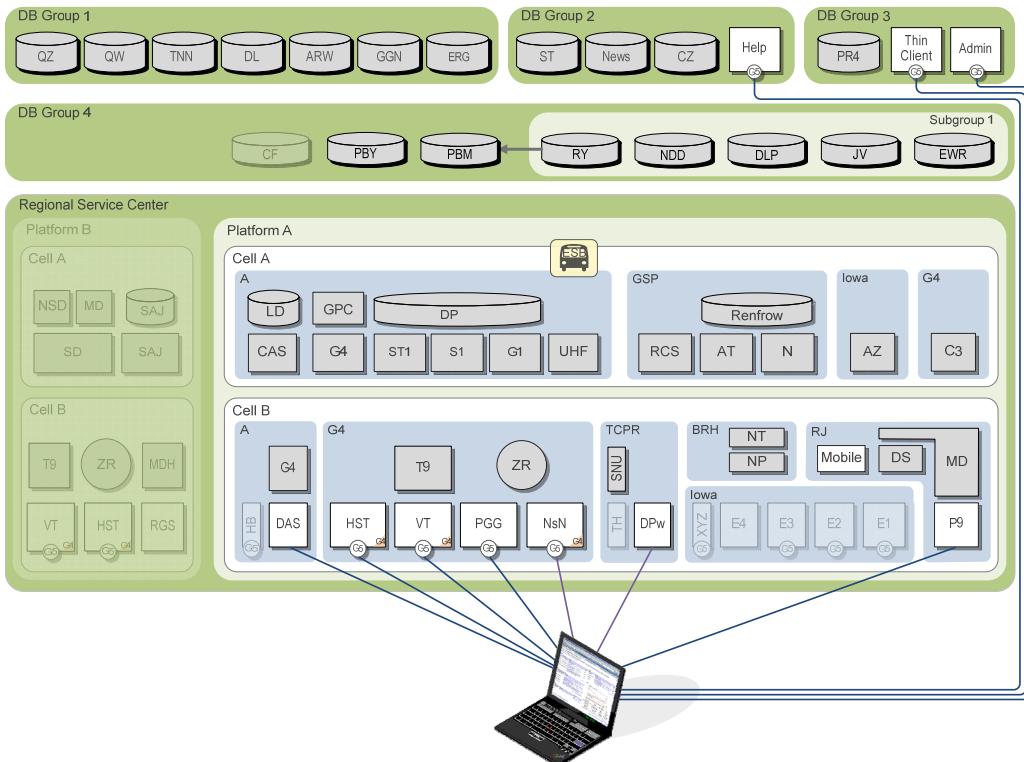
Jim Foley

Sample diagrams

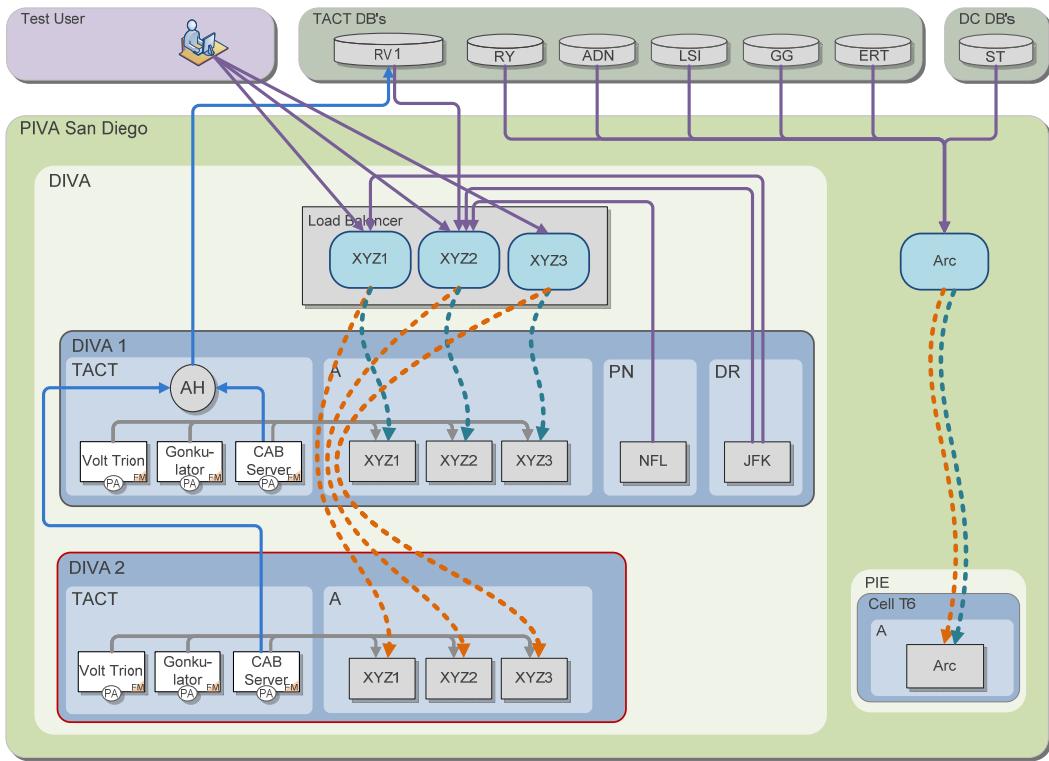
These Visio diagrams appeared in presentations, architecture documents, developer's guides, support training, etc. They were often used during conference calls to facilitate communication with upper management as well as with a diverse global group of several hundred contract developers. Details have been altered/omitted, management was shown this for permission, and no sensitive proprietary information is contained here.



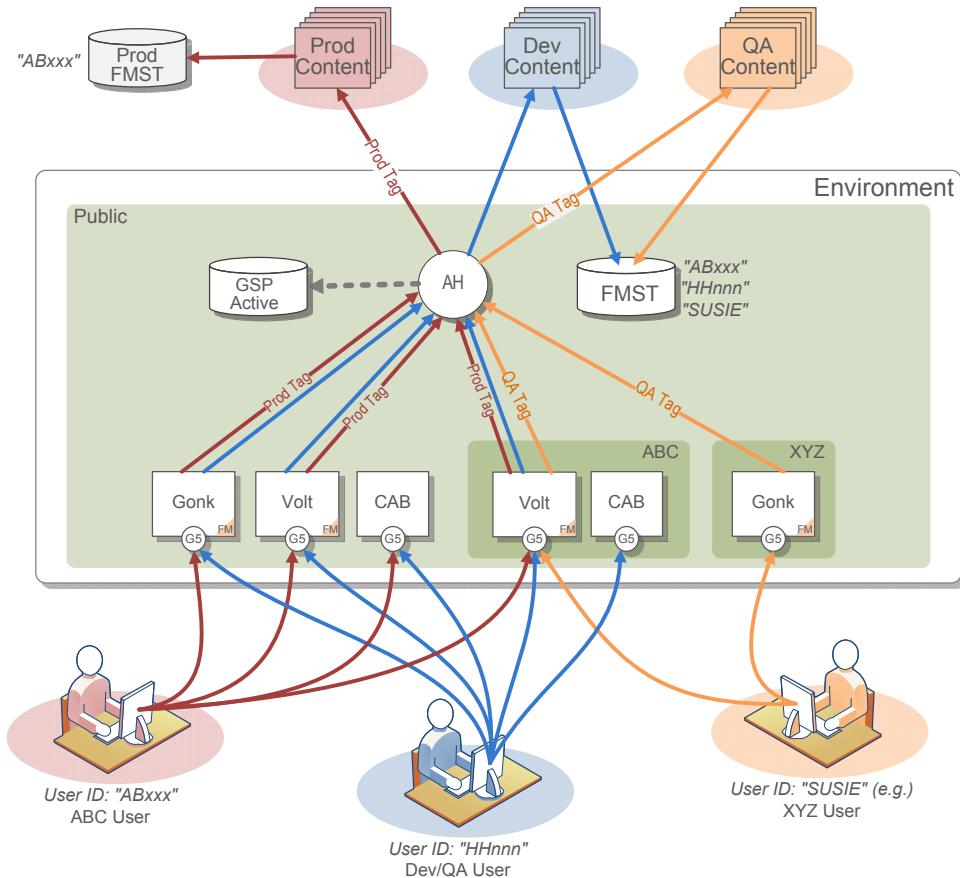
1. Conceptual view of various content & services feeding into different product lines & clients.



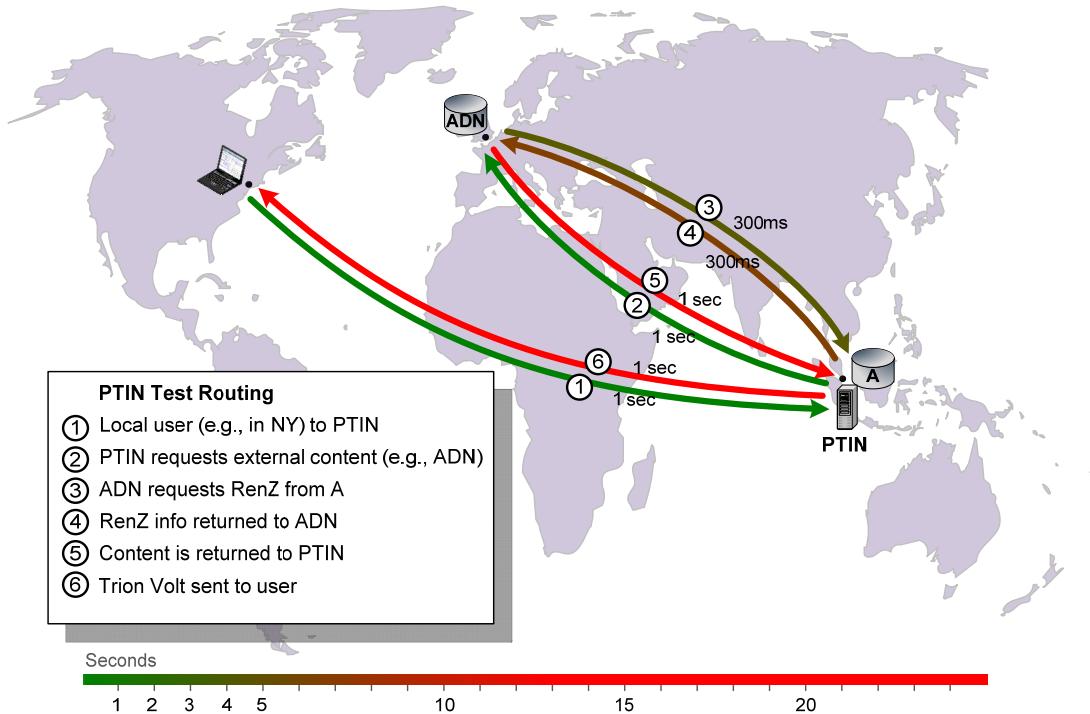
2. A web-based product architecture, showing different services located in different cells within the overall data center. Transparent areas show services used with other, related products. The yellow "bus" at center replicates certain data to similar data centers around the globe.



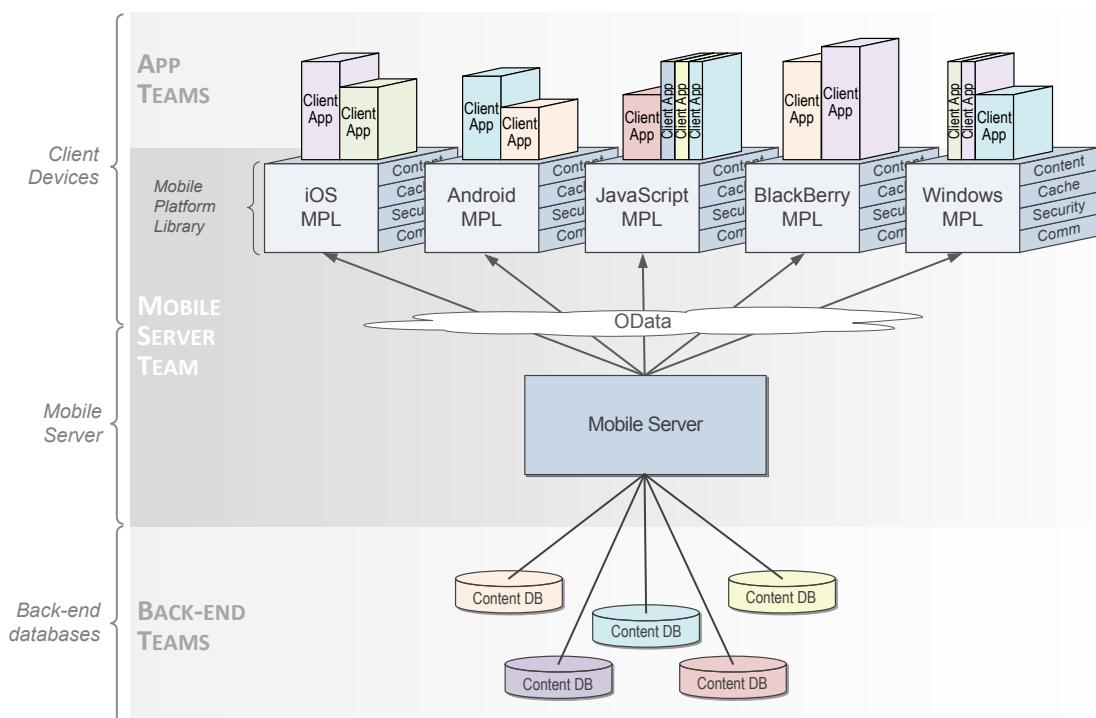
3. Two alternatives for connectivity in a test environment



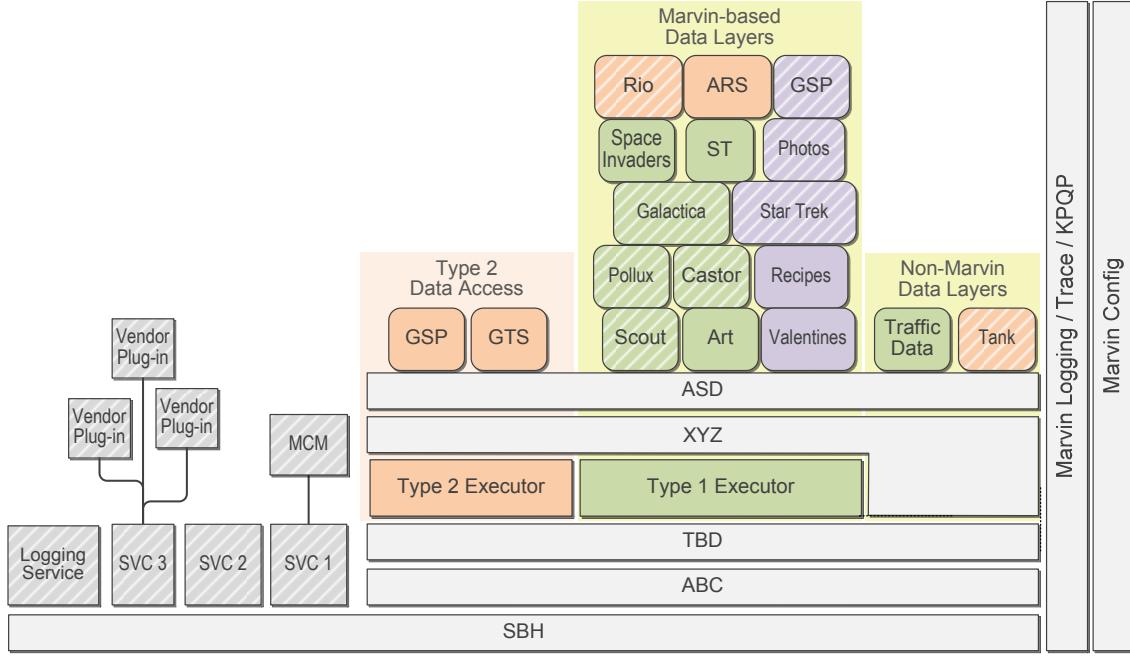
4. Getting the right content to three types of users, testers, and developers through a shared environment



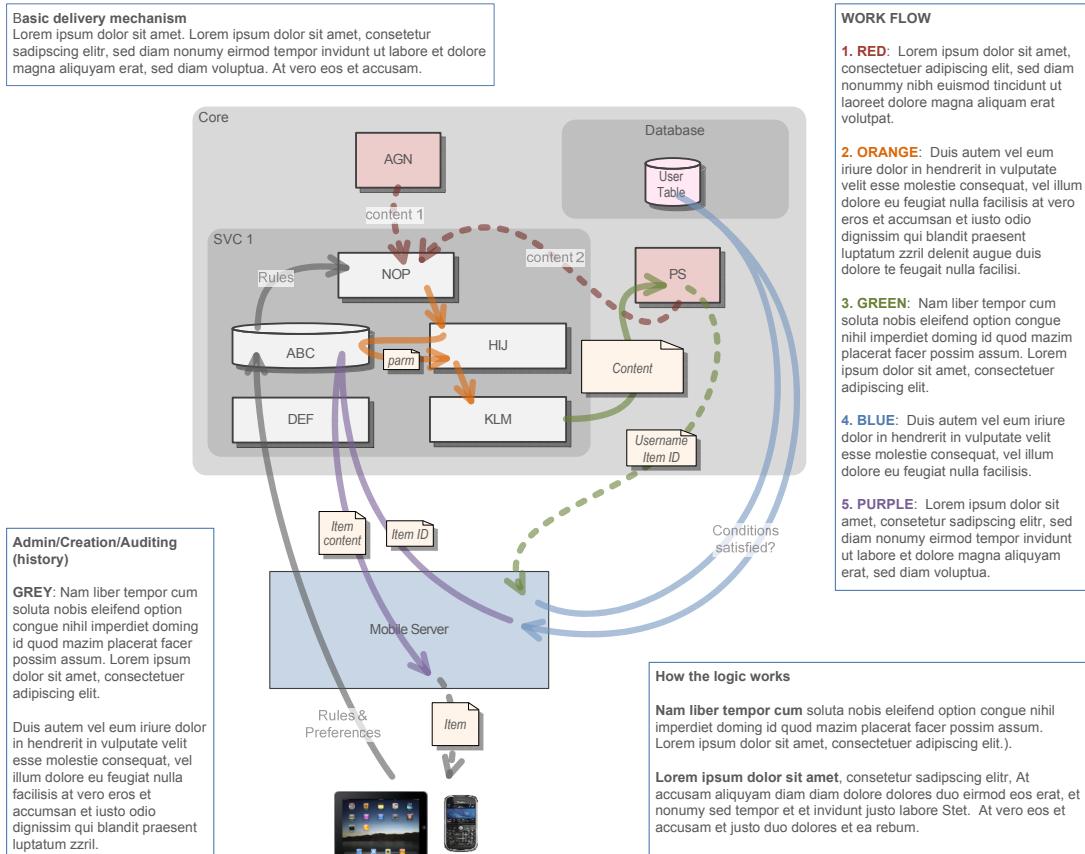
5. Explanation of why a particular test setup was giving slow response times – too many intercontinental hops. Colored timeline at bottom corresponds to the colors of the transit lines on the world map, moving from green (acceptable time) to red (unacceptable time) by the end of the fourth major leg.



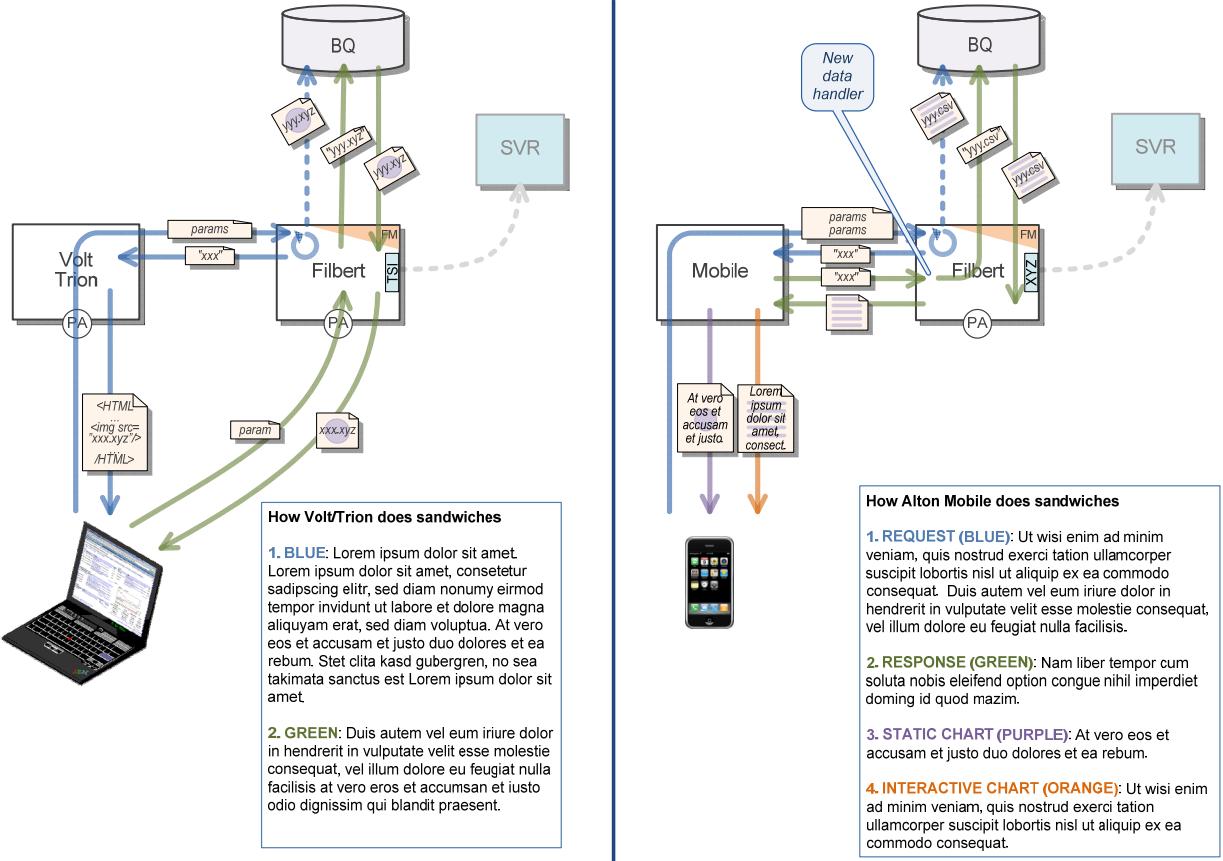
6. Product architecture concept for a proposed series of mobile products.



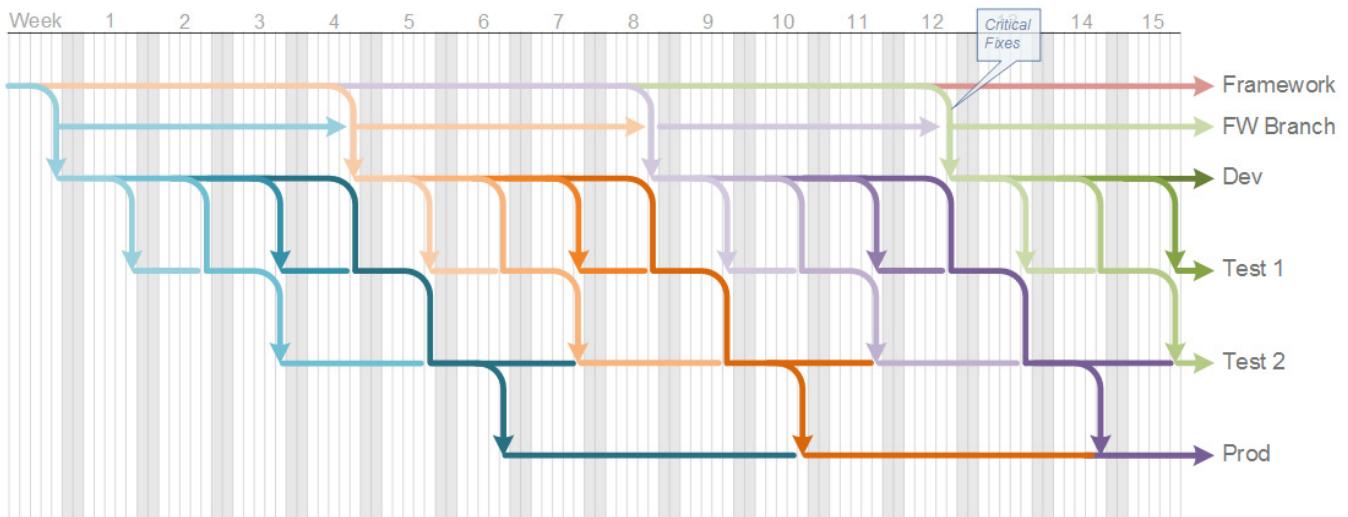
7. A software architecture for a proposed server, with the top layer (orange & yellow) accessing back-end content & services. Stripe patterns show which teams are responsible. Orange and yellow regions show the data protocol used by the components. Colored boxes show the different generations of the product, moving from purple and green (1.0) to green and orange (2.0).



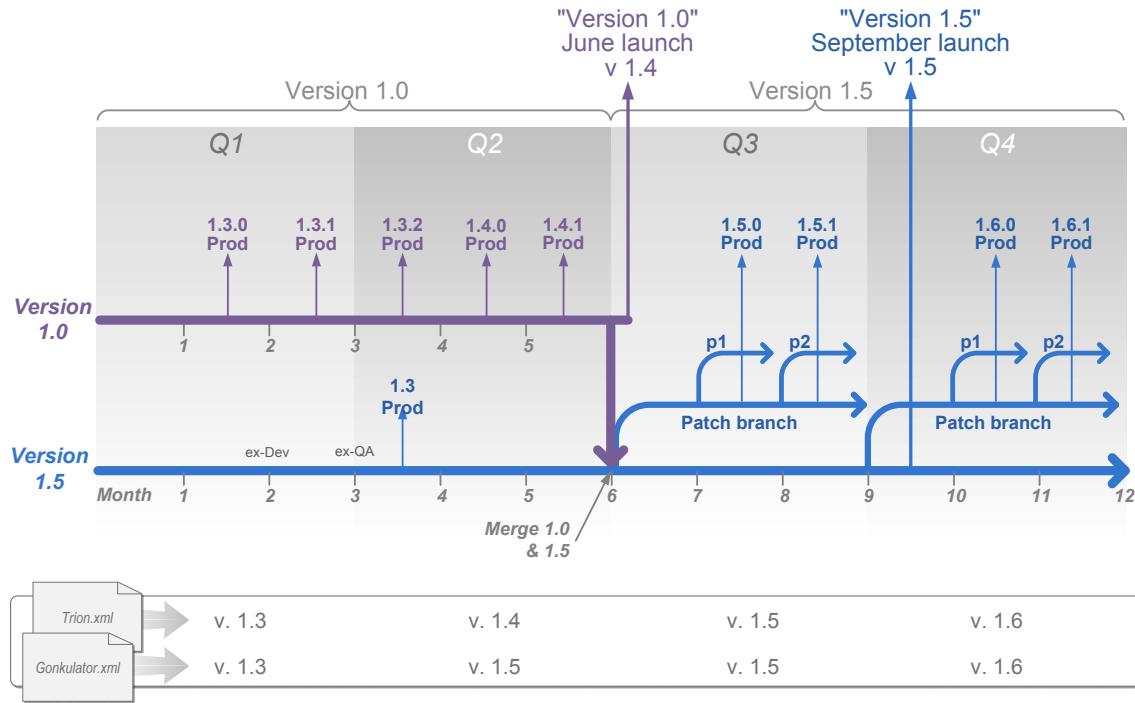
8. Explanation of workflow for a specific mobile user service



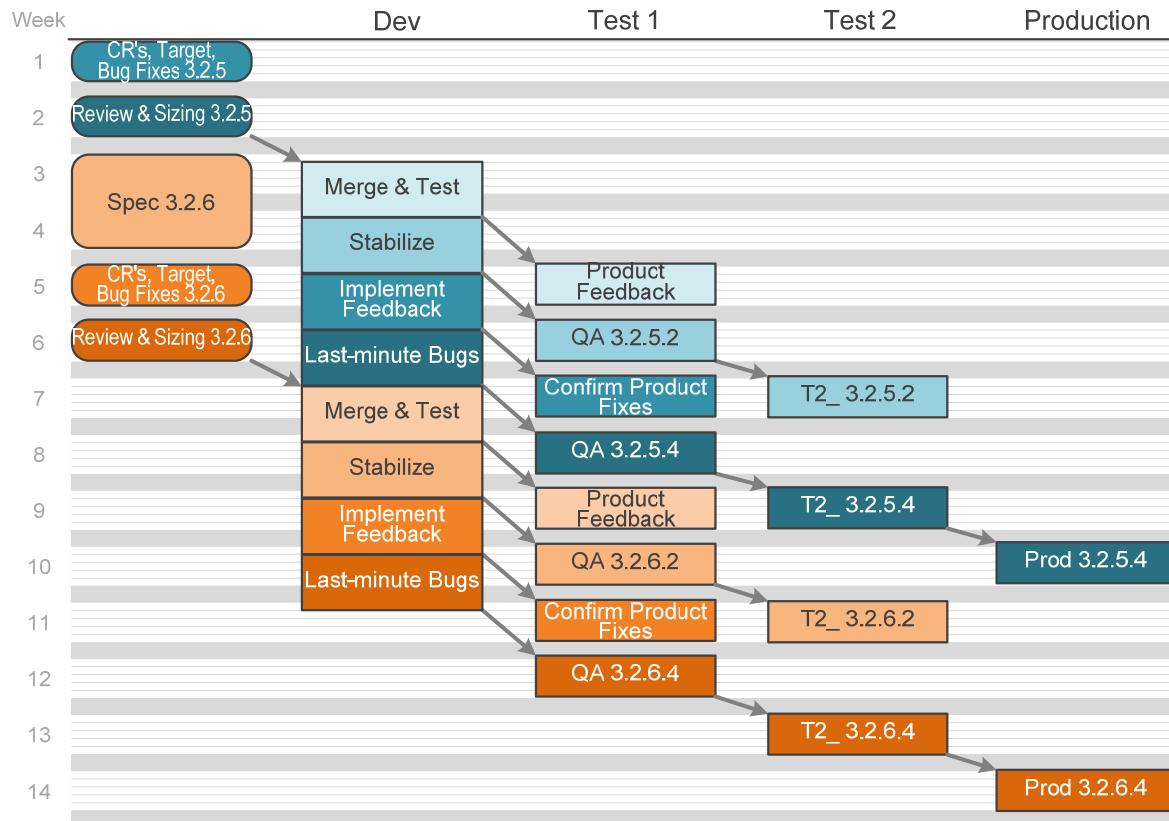
9. Comparison of analogous workflows between desktop server and mobile server



10. Conceptual view of code branches (colored lines) installed in various environments (horizontal rows, labeled at right) as the product evolves through monthly releases. Color shows sub-version (1.0, 1.1, etc.), shade is the progression of sub-sub-versions (1.0.0, 1.0.1, etc.), with the darkest shade indicating completion of that version. Weeks are numbered across the top; weekends are vertical gray bars. The framework code, being developed above, provides features needed for product development below and is integrated into the main branch every four weeks.



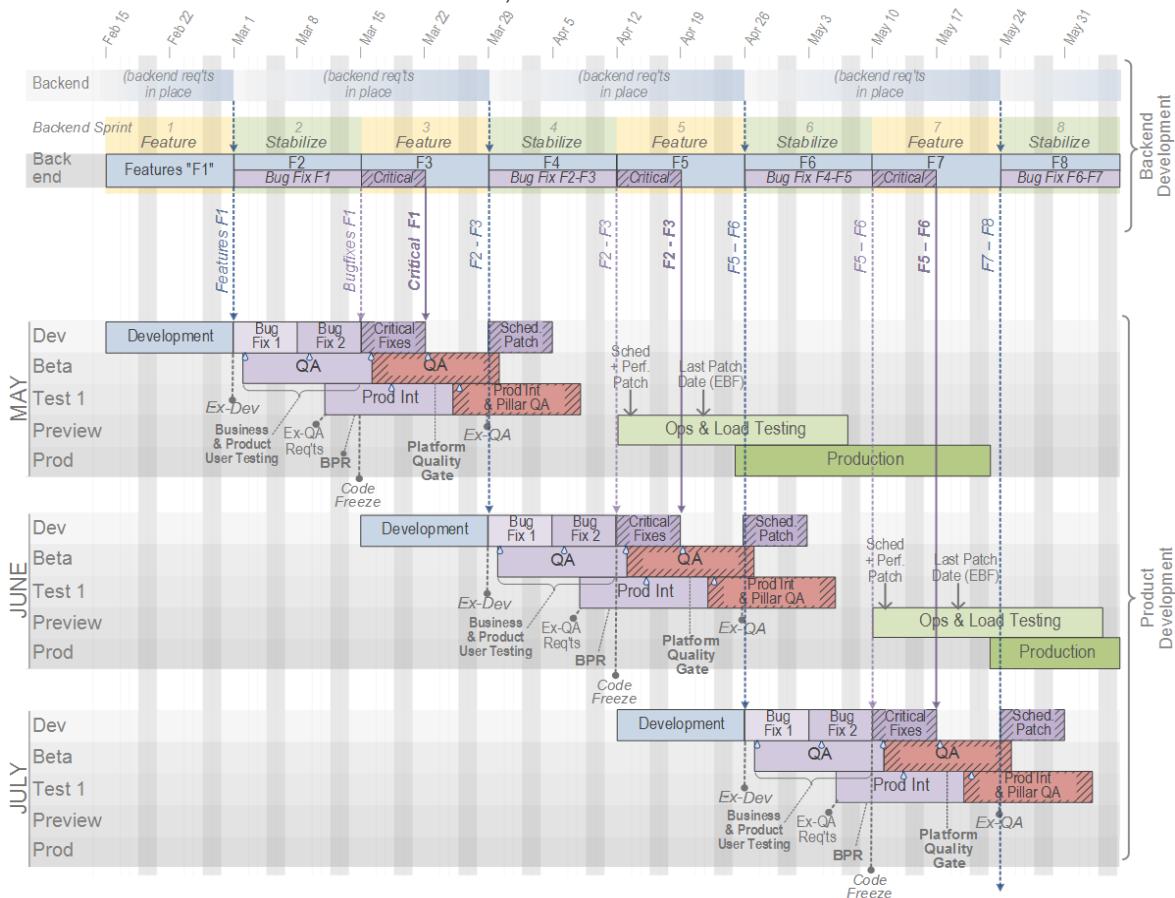
11. A plan for two major product versions being developed in parallel branches (1.0 in purple, 1.5 in blue), merging at the start of the third quarter. Upward arrows are releases to Production. At the bottom are the projected version numbers of the two major XML files that configure the product.



12. A sprint work plan, showing product versions under development moving through different environments in a 4-week cycle. Time is vertical, weeks numbered down the left. Color shows version, shade shows sub-versions approaching completion. Environments are columns. Every four weeks a version is released to Production at the far right.

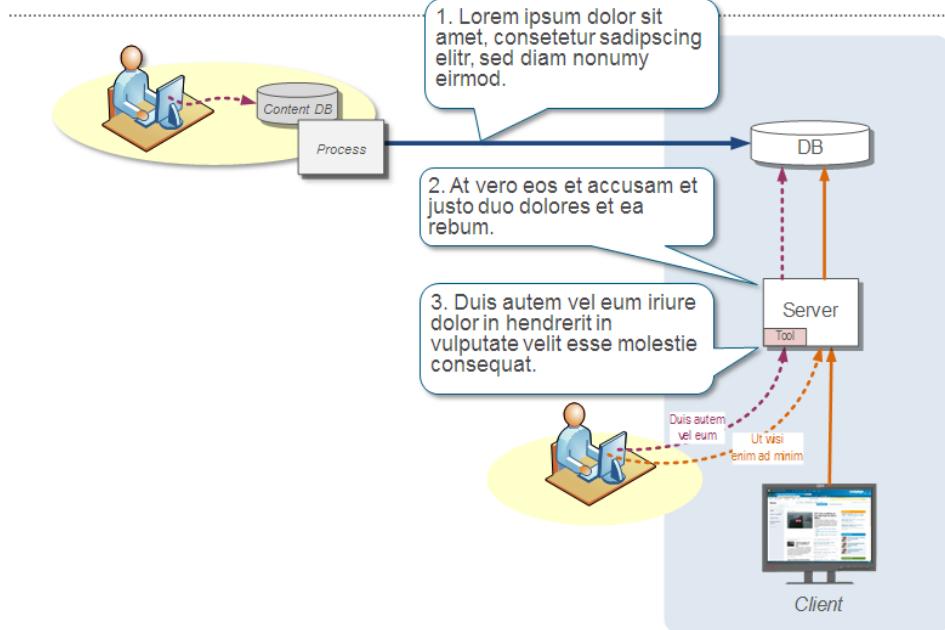


13. A 4-week sprint work plan, showing scrum activity in the leftmost three columns, and evolving versions installed in environments (blue columns) on the right. Weeks are numbered down the left. Color shows sub-version, shade shows sub-sub-version moving toward completion. Green is the monthly final version. At the top, two different sources serve QA content or production content to the various environments. A new product or version starts every four weeks, as a red arrow at the far left.



14. A highly-detailed development cycle, taking 10 weeks to implement a feature from start to production. Dates go left-to-right across the top; vertical gray bars are weekends. Colors are stages – Dev, QA, operations testing, critical bug fixes – with Production in green. Horizontal gray shaded bars are five environments where the developing product is installed during the various stages. The four-week News back-end development timeline, shown above, provides required new features F1 through F8 for the user-product development below.

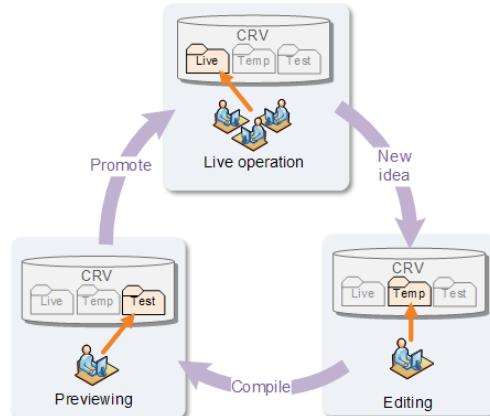
Phases



15. A slide explaining a work process.

Editing CRV files

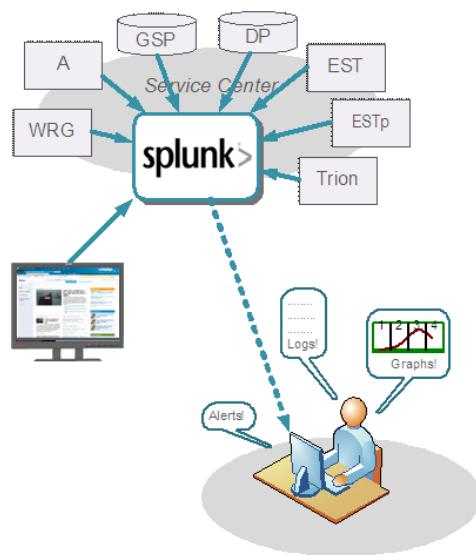
- Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit. lobortis nisl ut aliquip ex ea commodo consequat.
- Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.
- Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.



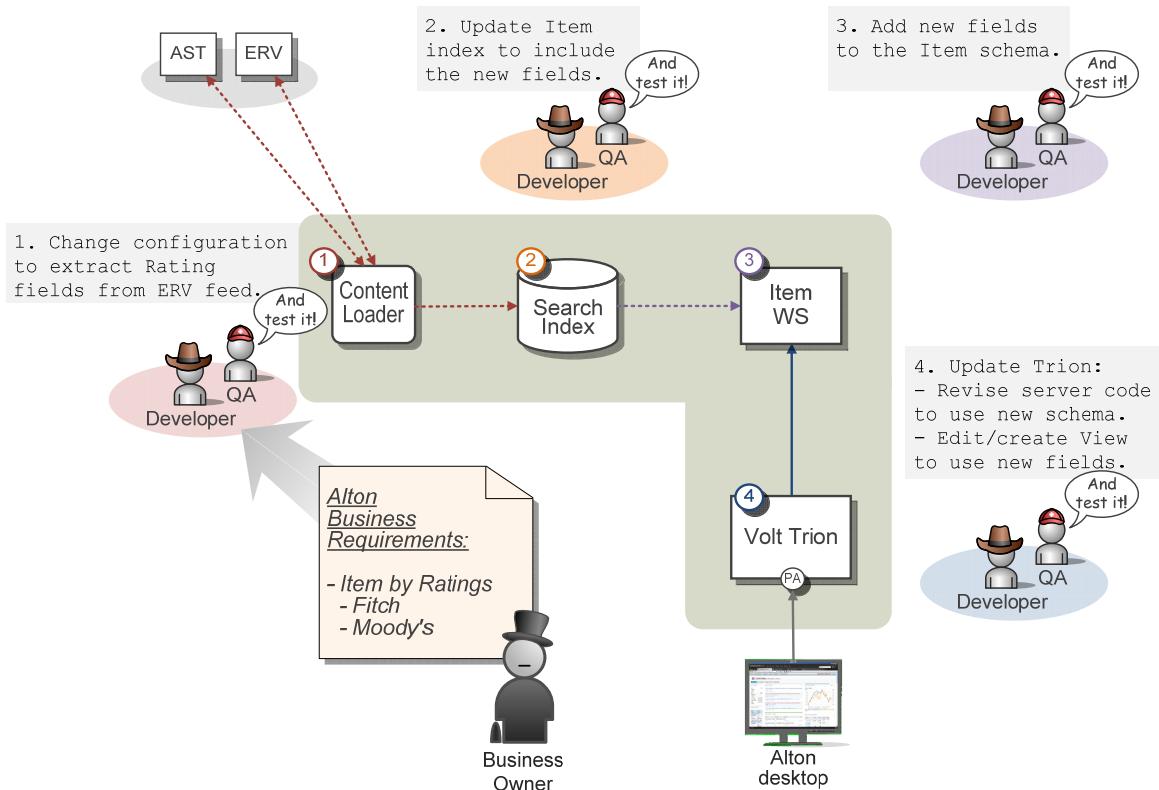
16. A slide explaining a work process.

Splunk

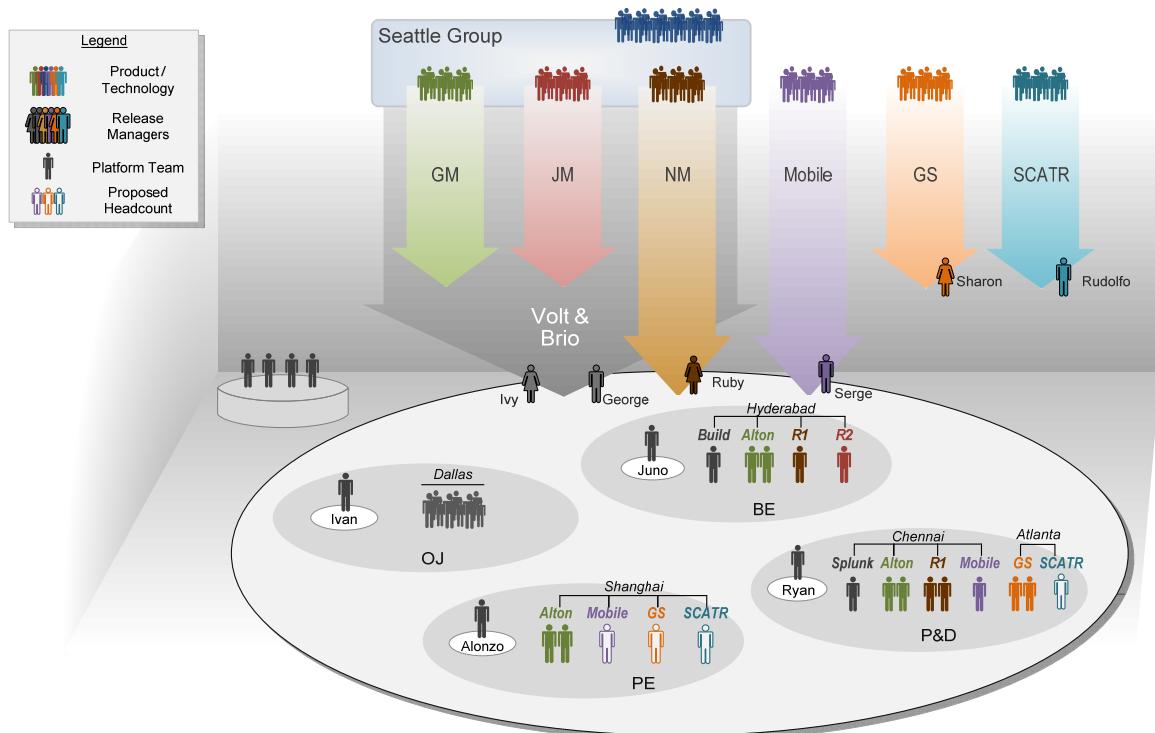
- Splunk LightWeight Forwarder (LWF) is a 3rd-party product that allows fast, convenient access to system-wide logs, including server logs and user application logs.
- It forwards application logs and event logs in near-realtime to a Splunk Index server within a service site.
- Logs and log-related data can be accessed at the Splunk Index server.



17. A slide explaining Splunk, a tool that gathers system performance metrics.



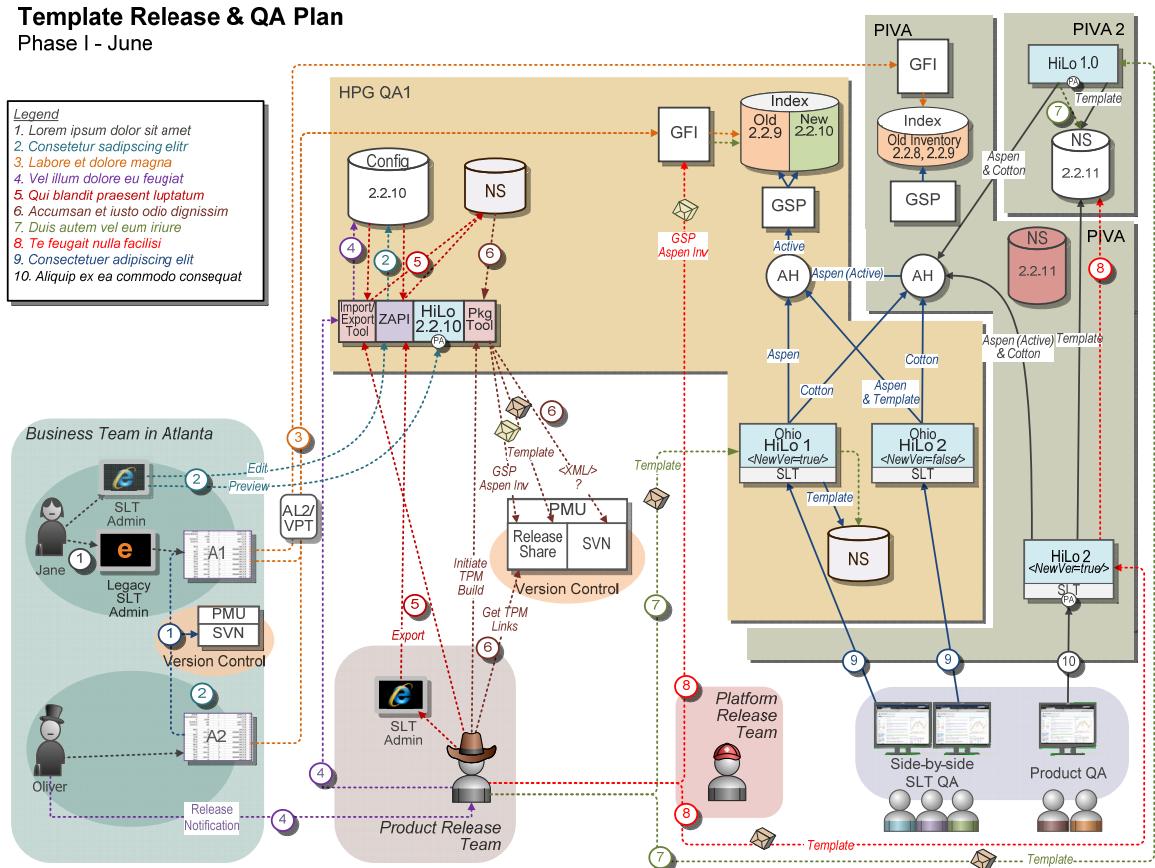
18. A diagram explaining a work process. Hats indicate roles, colors indicate stages of the process.



19. A village diagram, used to discuss roles and interactions between various teams and workflows.

Template Release & QA Plan

Phase I - June



20. A busy village diagram used on a series of calls to discuss workflow for editing, distributing, and testing metadata – one of four diagrams that changed daily for a week. A few key elements have been removed to protect sensitive information. Shaded areas are teams, shadowed rectangular regions are environments, line colors show numbered stages in the process, line styles are varied for clarity and to distinguish work flow from actual data requests. (Imagine these calls without this diagram!)