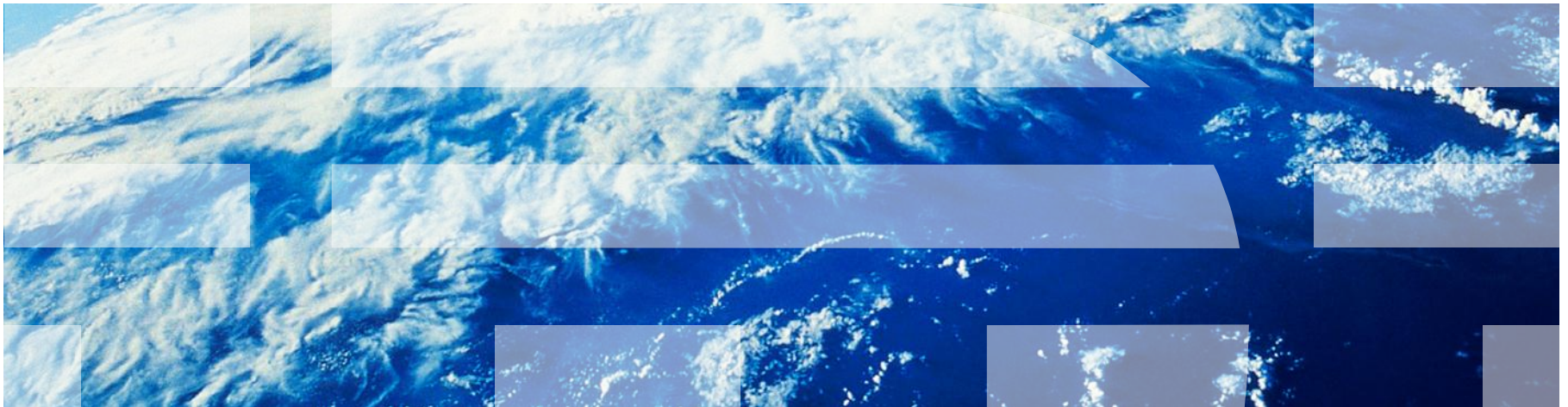


# Any Record, Any Place, Any Time

Open Services Lifecycle Collaboration (OSLC) Application Development





## Presenter Bios

### **Pete Steinfeld**

- Pete has worked for Rational since it merged with Verdex Corporation in 1994. He's worked on embedded systems, debuggers, compilers, build systems, and, most recently, on the Rational OSLC Client Request (ROCR) system.
- Pete is spending his children's inheritance on woodworking tools.

### **Bob Myers**

- In the last 14 years, Bob has designed 2 production Client Request systems for the IBM Rational Brand using Rational's software development Tracking tools
  - RATL (single system) and RATLC (distributed system) based on ClearQuest and ClearCase
  - Rational OSLC Client Request (ROCR) based on OSLC, ClearQuest, RTC, Change, Insight and Omnifind
- Bob has collaborated with a few talented and imaginative engineers to produce a patented Composite State Management system, CQ ALM, used, in at least one case, to track integrated hardware and software changes in the development of the iPhone and iPad chips
- Bob is a beekeeper and Ocean shell rower



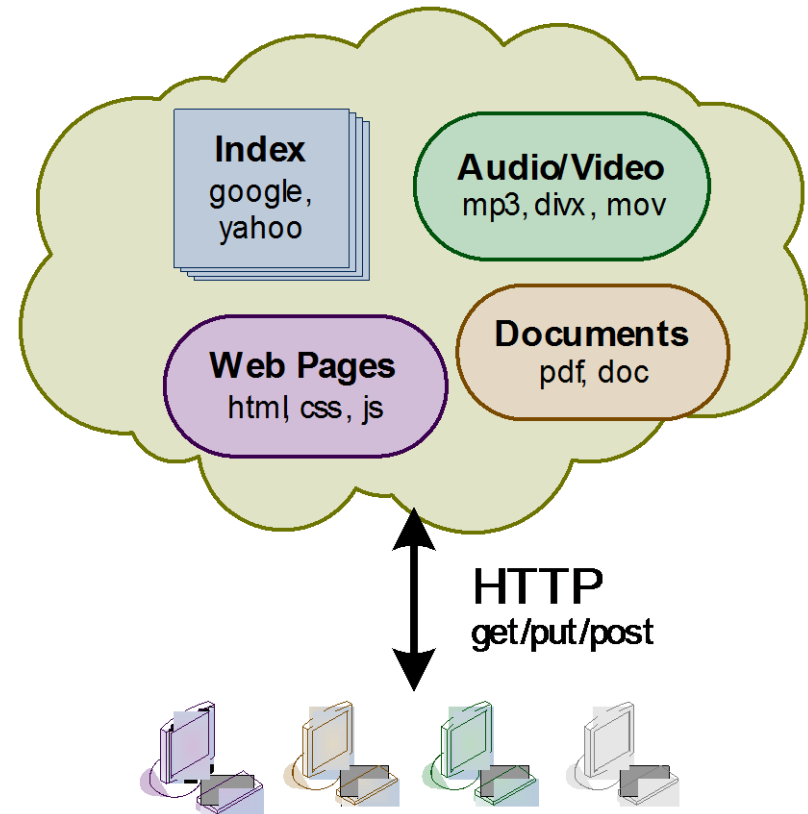
## Agenda

- **OSLC overview**
- **Modular development**
  - Parallel, independent, **Component-based** Development
- **Sync replication is limited or eliminated**
  - Limited data duplication
- **Data Warehouse and Indexed Search**
  - Federated reporting and search
- **User experience**
  - UI Preview limits context switching
- **OSLC Standard set**
  - **Optional OSLC support that help if implemented**
- **Wrap-up**
  - Questions and discussion



## The Internet – an inspiration for an architecture

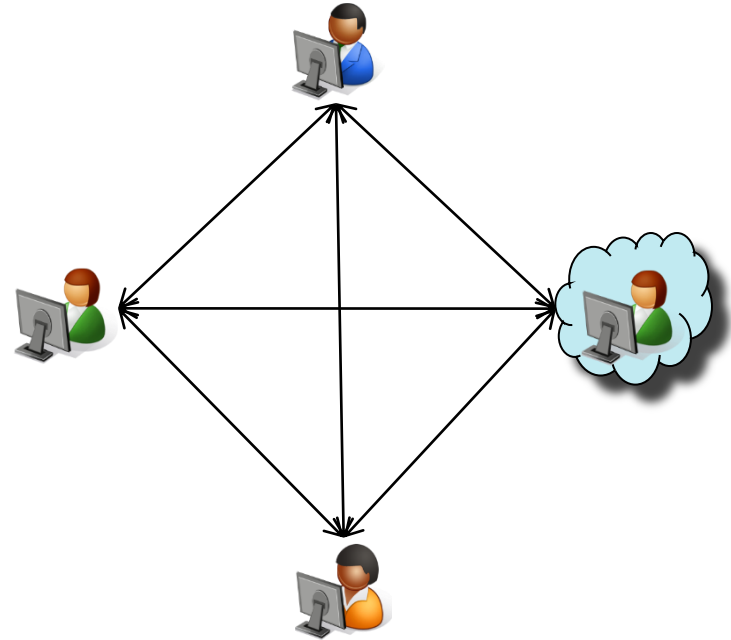
- Amazingly scalable
- Integrates information on a massive scale
- Infinitely extensible
- Collaboration on unprecedented scale
- Open
- World-wide information visibility
- Unprecedented business opportunities





## Traditional Tool Integration. Ouch.

- $N^2$  possible point-to-point connections
  - Limited coverage
- Closed APIs
  - Vendor lock-in
- Tight Coupling
  - Dependence on internal structures and meanings
- Lockstep upgrades
  - Version incompatibilities





# What Makes the OSLC Approach Better?

## Traditional Approach

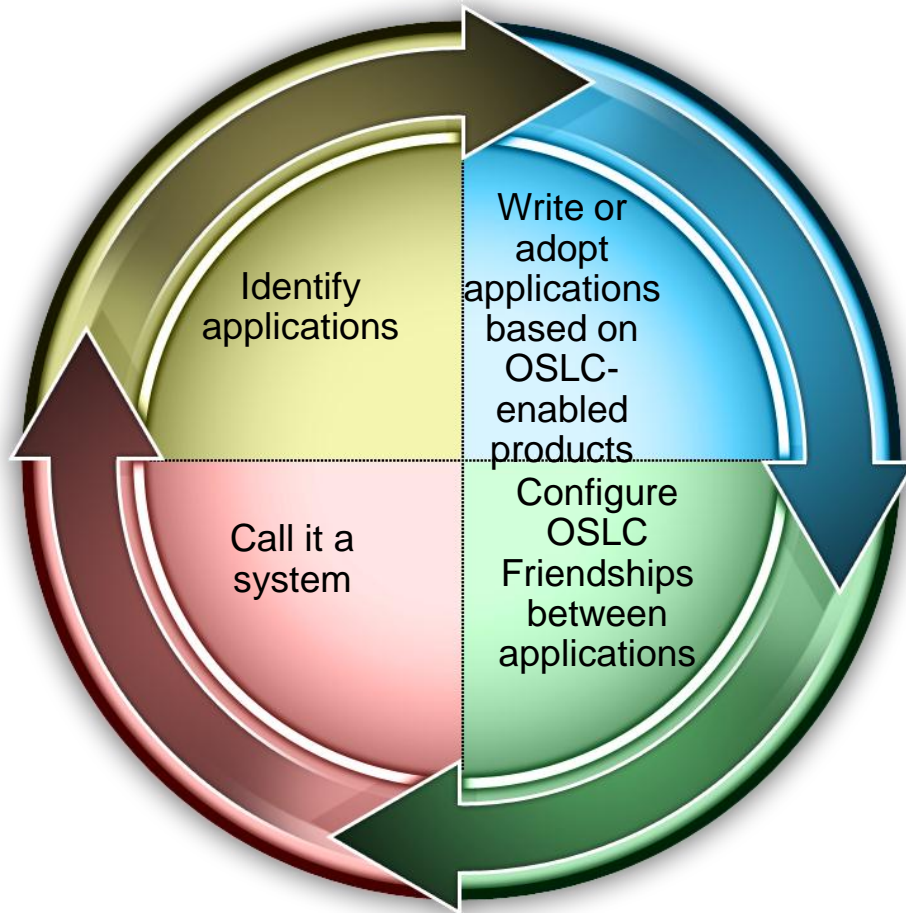
- Brittle integrations, version-specific APIs
- Monolithic repository or import/export
- “Boil the ocean” meta-model design
- Forced migration to a common code base
- Premature architectural decisions
- A vendor-led “partners” program
- Duplication of capability

## OSLC Approach

- Loosely-coupled
- URLs
- Minimalist
- Technology-neutral
- Incremental
- Open
- Delegation of capability



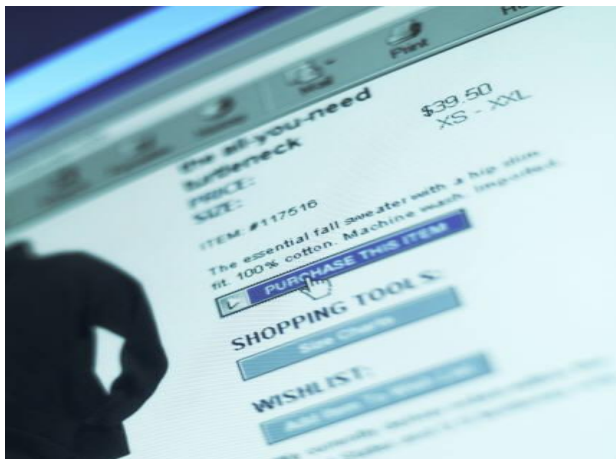
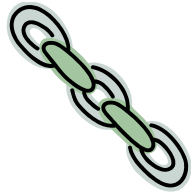
# Agile OSLC-enabled System Development



- Minimalist/additive approach
  - No need to have a comprehensive data model with standardized field and record semantics
- Do not need IT Admin professional to comprehend whole application from a coding, record layout, field naming, standardization perspective. (100 page white papers)
- Design solution is by Business Analysts at the business component process level using OSLC-enabled applications



## Modular (Component-based) development



- Adoption of parts of a system rather than all or nothing
- Maintenance is limited to one application, not the whole system
- Access is controlled at the Application level and Single Sign On (SSO) technology makes Connects invisible to permitted users
- Employs tools that are better suited or more accepted rather than needing to adopt one tool for everything
- Scales easier because you can access any record, any place, any time for OSLC enabled applications
- Supports including new applications without rewriting the original system
- Promotes independent, parallel development

In our case, each application had been vetted in production. independently for more than a year by hundreds of engineers, 24X7, around the world before we deployed





## Data Warehouse and Indexed Search

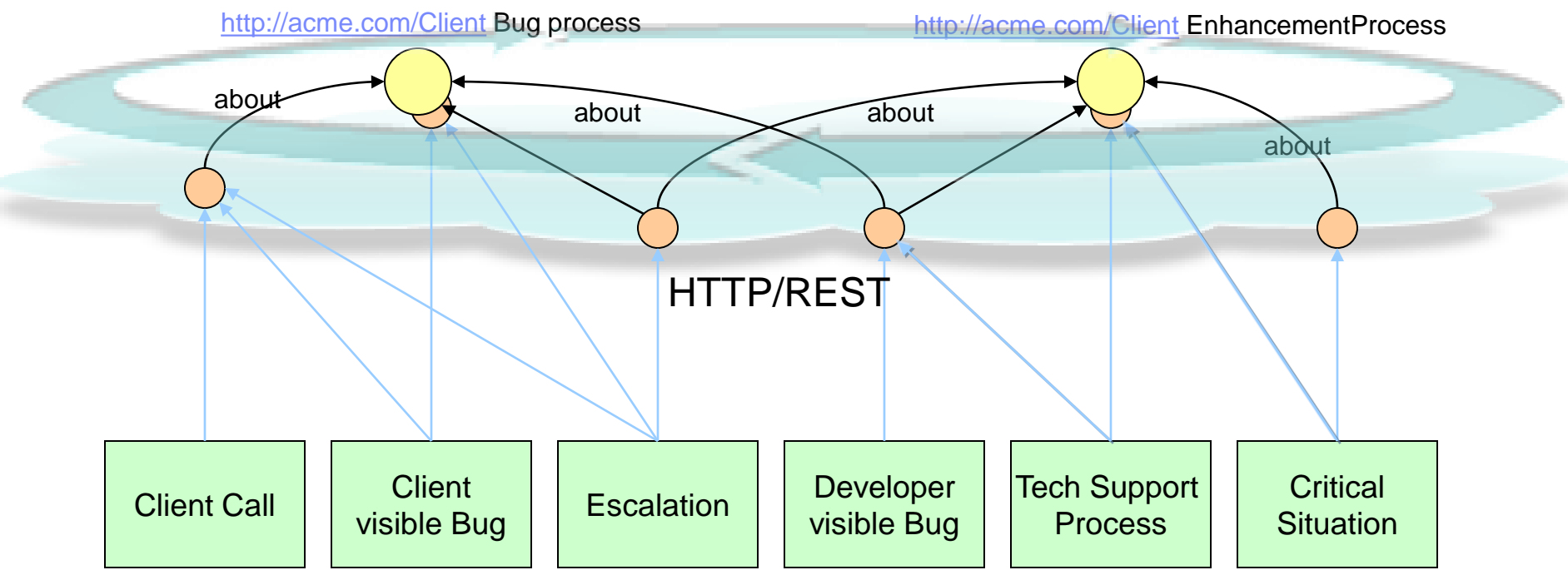
- In a Data Warehouse
  - OSLC makes linked records look like one extended record for application reporting
  - Queries can filter based on Type of Link
  - Information filtering involving data from different OSLC-Linked applications
  - Records in any OSLC-enabled application can be linked
  - IBM Analytics can be used to make sense of this data

Indexing the Data Warehouse provides a faceted, string-based search capability





# Data Integration – the new way – “www linked data”





# User experience and Scalability

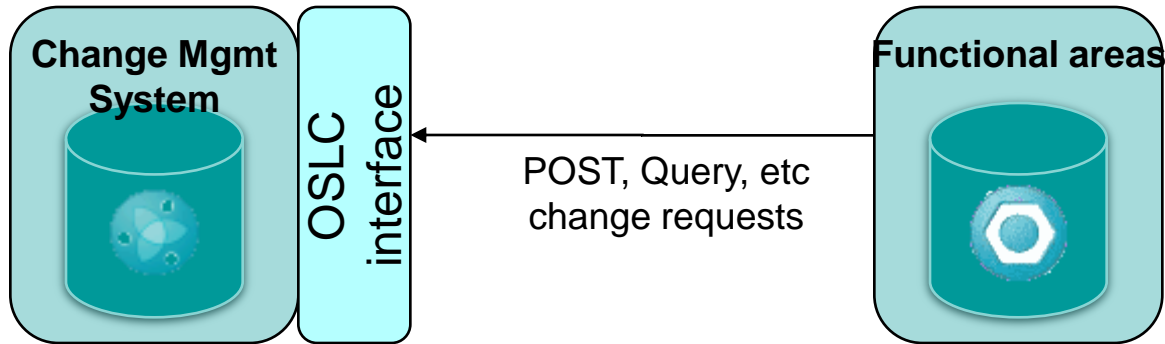
- **User Experience**
  - Context switching is reduced by OSLC UI Preview so the differences in look and feel between independently designed and developed applications is reduced
- **Scalability**
  - Due to modular security, linking can extend to other teams and other companies





## OSLC Suggested Standard Application Support Set

*Loosely coupled integration with “just enough” standardization*



- Able to Link (make Friends) of each OSLC-enabled application
- Able to Link when creating or selecting records in each OSLC Friend
- Able to have a hover UI Preview on all linked records
- Support OSLC API sufficiently to
  - Enable pulling OSLC Linking from OSLC-enabled application repository into a Data Warehouse
  - Pre-fill of data upon OSLC Link-create
  - Governance support like Parent-Child State correlations

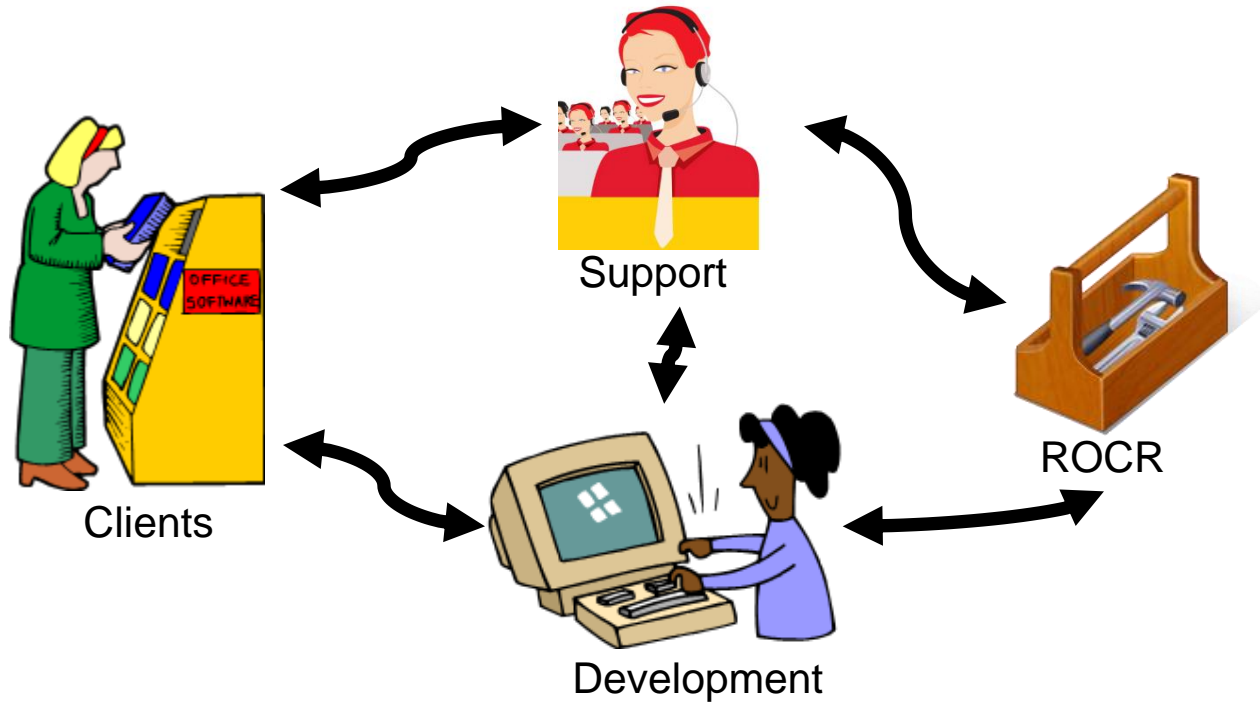


# ROCR Demo



## What is Rational OSLC Client Request (ROCR)?

ROCR is a toolkit for Rational Client Support and Development to collaborate with clients and with each other





## What is Rational OSLC Client Request (ROCR)?

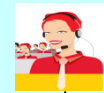
### ▪ Client interactions –

- Problem management reports (PMRs)
- Authorized Program Analysis Reports (APARs)
- Requests for enhancements (RFEs)
- CritSits (Customer Satisfaction Project Office or CSPO)



### ▪ Collaboration between Support and Development

- Escalations
- Development work items



### ▪ Rational Client Support dashboard

- Central point of access



## Operating environment for Support and Development

These are the tools that we cannot change and that ROCR must interact with

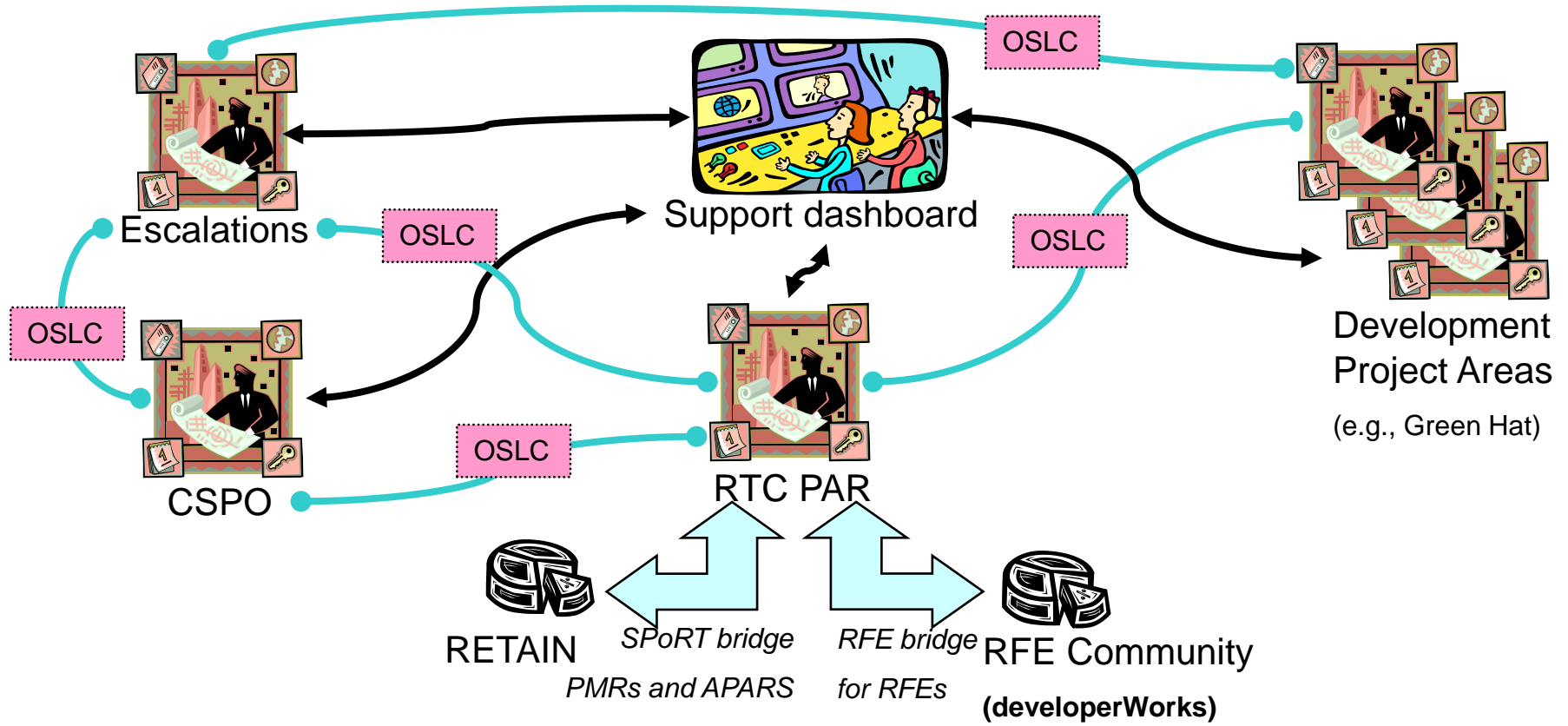
- RETAIN (Remote Technical Assistance Information Network) database for PMRs and APARs
- RFE Community
- Development work item and change request systems --
  - Many Rational Team Concert (RTC) project areas used by development teams
  - RATLC used by some teams
  - Telelogic Change used by some teams







# ROCR overview





## ROCR RTC project area summary

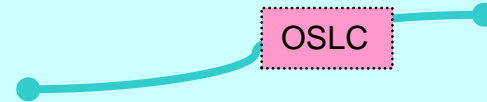
- Existing development project areas (Green Hat, RAD, etc.)
- [Escalation project area](#)
  - Owned by Rational Client Support
  - Holds the Escalation work item type
  - The [“All Products” dashboard](#) with navigation, help, query widgets, etc.
- [RTC PAR \(PMR, APAR, RFE\) project area](#)
  - Bridge to RETAIN for PMRs and APARs
  - Bridge to developerWorks for RFEs
- [CSPO \(Customer Satisfaction Project Office\) project area](#)
  - Used for CritSits
  - Holds the Situation work item type





## OSLC linking capabilities

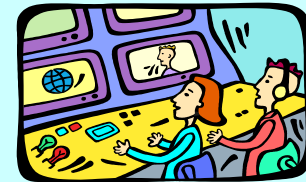
- Enables direct navigation between applications
- Rich hover
- Automatic cross link creation
- Access to capabilities of other application
  - Work item creation
  - Work item search
  - Query execution





## The “All Products” dashboard

- Central point of navigation for all ROCR functionality
- Lives in the Escalation project area
- Has tabs base on role – L2, L3, SWAT, ...
- “Navigation and Help” tab contains “How to’s”
  - Create Escalation
  - Triage an APAR
  - ...
- Work item widgets reference queries
  - Referenced queries can be in other project areas (Escalation is in Escalation, PMR/APAR are in RTC PAR, etc)





## Infrastructure needed to support ROCR

- ROCR System Architecture
- RTC project areas for Escalation, PAR, and CSPO
  - Servers
  - Databases
  - Friendships between the project areas
- Servers to run RETAIN (SPoRT) and RFE bridges



Questions?



Thank  
YOU







## Abstract

- We have deployed a set of five applications developed using different OSLC-enabled software products, created independently by business specialists from different divisions within our company. Each application was independently vetted in a production environment by hundreds of engineers around the world using them on a 24 hour X 7 day basis for more than a year. Each application has its own security facilitated by Single Sign-On technology.
- Application context switching is minimized by mouse hover and UI preview of OSLC linked records
- This loosely-coupled system replaces a tightly-coupled, synchronization-replicating application whose design required interdisciplinary business knowledge and code needing many conditions
- Synchronization-replication of data across applications has been minimized. Duplication has been reduced to only the data that does not change and needs real-time reporting.
- Records are extracted from various application databases and transformed into standard format. Once loaded into a Data Warehouse, OSLC-linked records are reported as though they are one record.
- An index database created from the Data Warehouse provides system-wide, string-based, text searching