

# IBM Rational Software Development Conference 2008

WHERE TEAMS ARE **R-HEROES**



## SDP30: Jazzing Up IT

Best Practices from Internal Deployment of IBM Rational Team Concert

**Michael O'Rourke**

Vice President, Rational Software Development  
mikeorou@us.ibm.com

# Agenda

- Goal of the presentation
- Centralized deployment of Rational Team Concert
- A Simple RTC Configuration
- Considerations for deploying Rational Team Concert
- Rational Software's RTC Deployment
- Benefits, Best Practices, and Lessons Learned

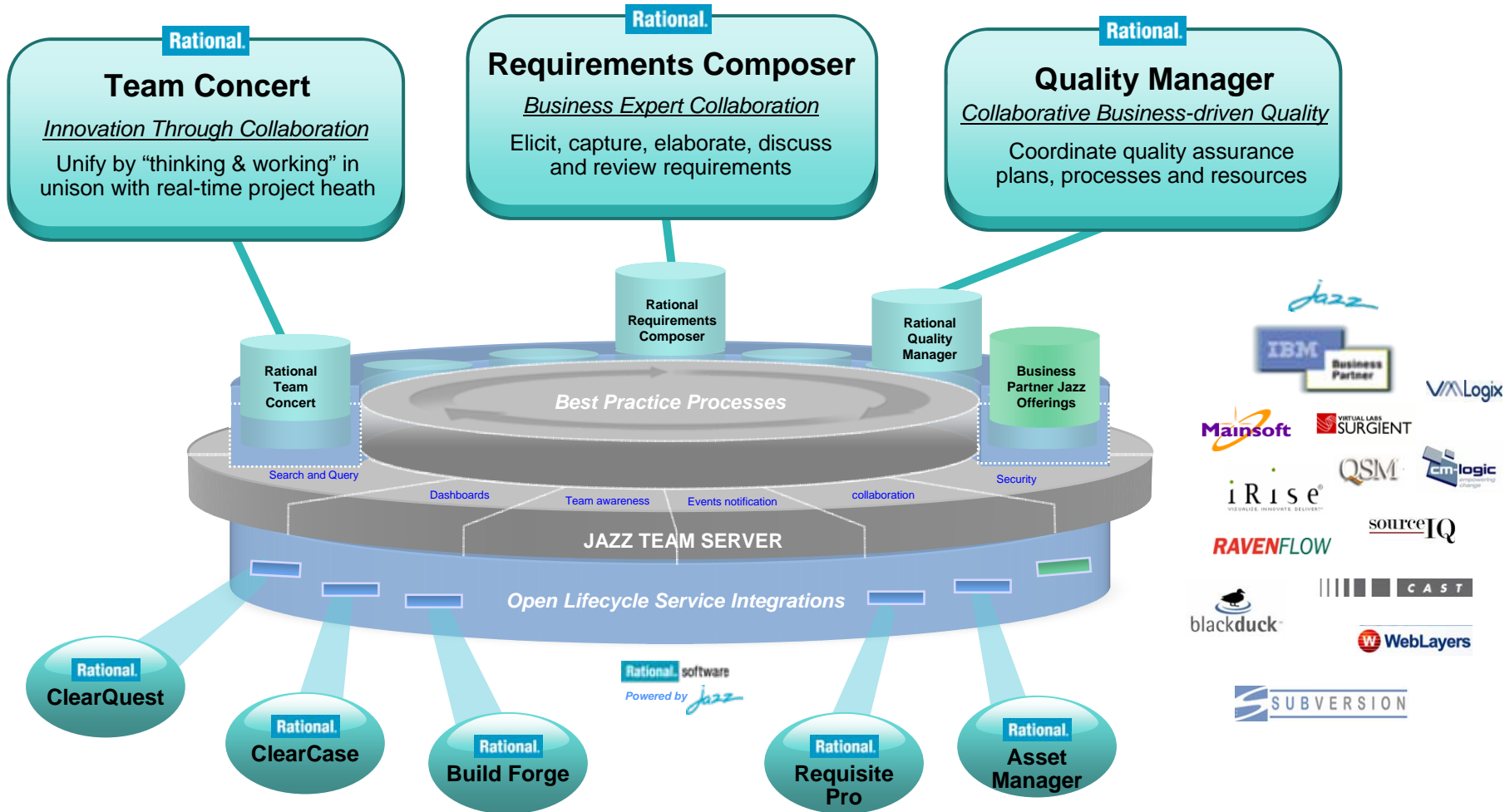


## Goal of the Presentation

- Help you think about how to deploy Rational Team Concert
- Give you some suggestions as to possible architectures and configurations
- Discuss some benefits, best practices, pitfalls, and lessons learned



# Introducing the first wave of new *Jazz* offerings



# Agenda

- Goal of presentation
- **Centralized deployment of Rational Team Concert**
- A Simple RTC Configuration
- Considerations for deploying Rational Team Concert
- Rational Software's RTC Deployment
- Benefits, Best Practices, and Lessons Learned



# “Jazz” adoption was rampant at Rational!

Easy to deploy and use

Minimal server requirements

Downloads  
Download freely accessible

Grassroot movement to adopt new technology



Collaborative



Agile process support

eclipse -based

“Drink your own champagne”



Support for distributed development



Developer friendly, team friendly



# Pros and cons of grassroots adoption of RTC

## PROS



- People get a feel for how easy it is to set up and administer
- Great as a sandbox learning environment
- Appropriate for workitems or non-source-code assets
- Control over process and workflow
- Fewer restrictions in security
- Control over upgrade schedule

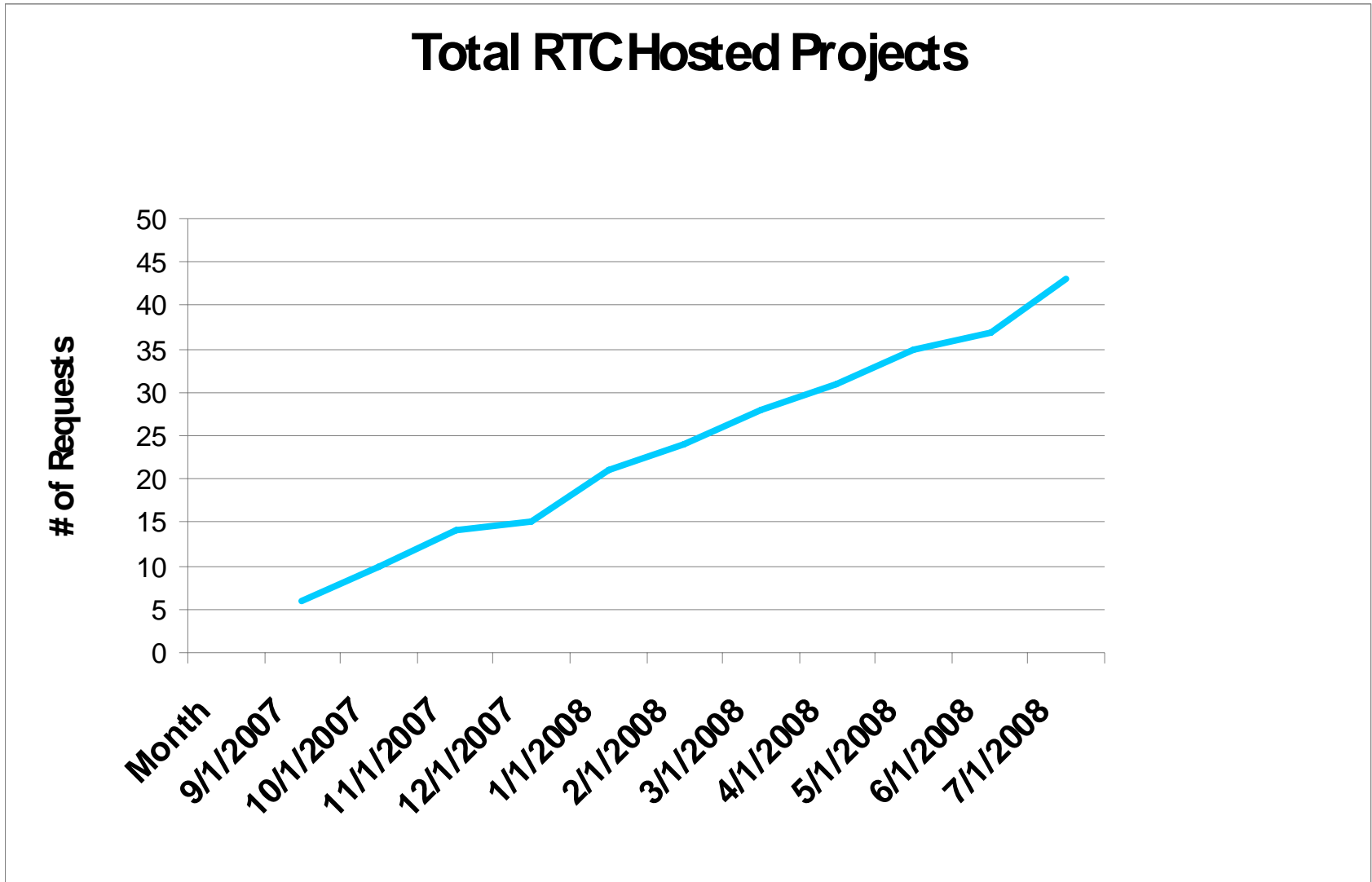
## CONS



- Isolated islands of data
- Lack of security and business controls
- Harder to impose standards
- Harder to develop continuity plans
- Variance in versions of SW across teams
- Lack of centralized administration puts burden on owning team – backup and restore, 7x24 support, etc.
- No economy of scale on hardware
- Decentralization of support requests

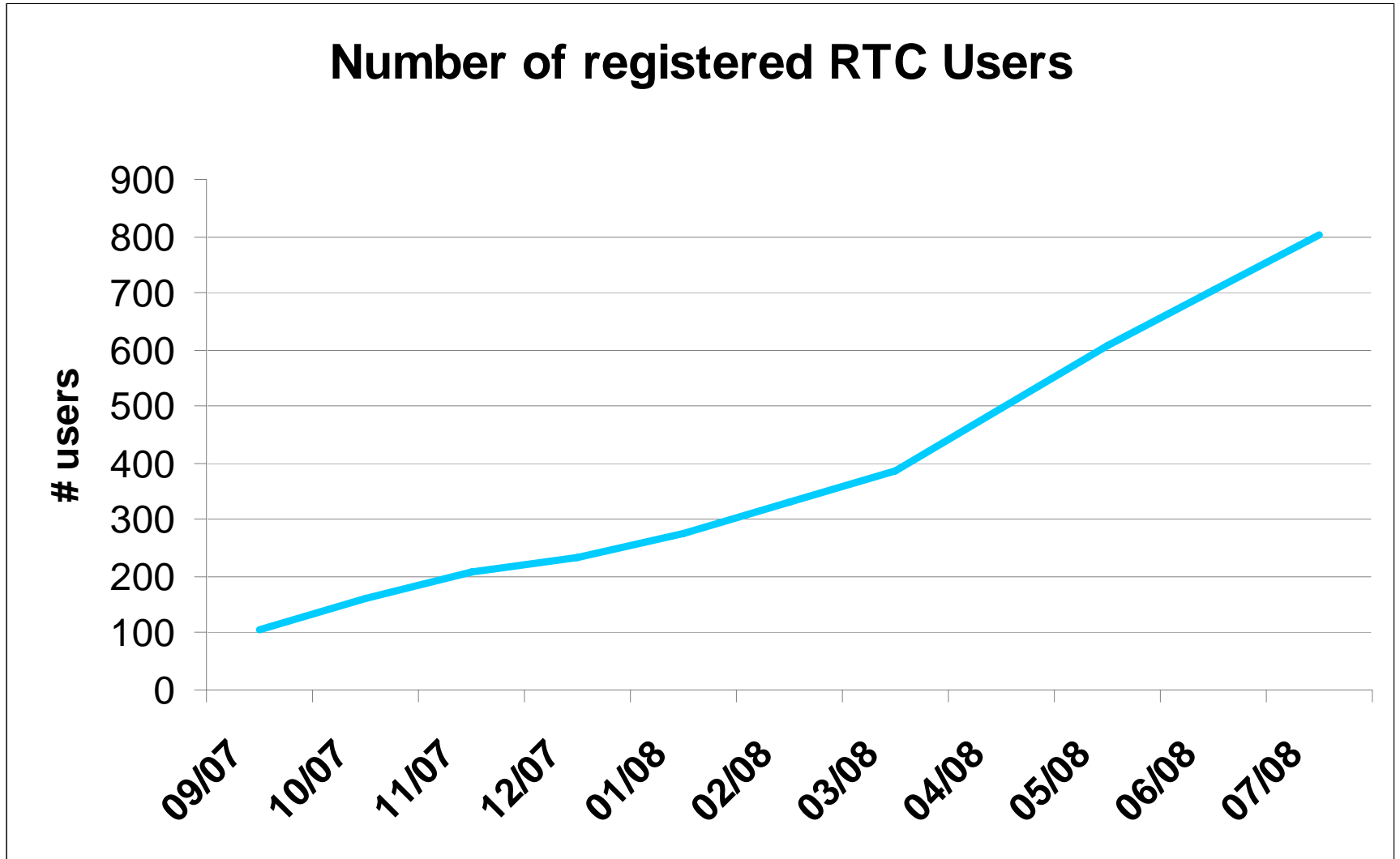


# Project Growth Rate at Rational





# RTC Adoption at Rational

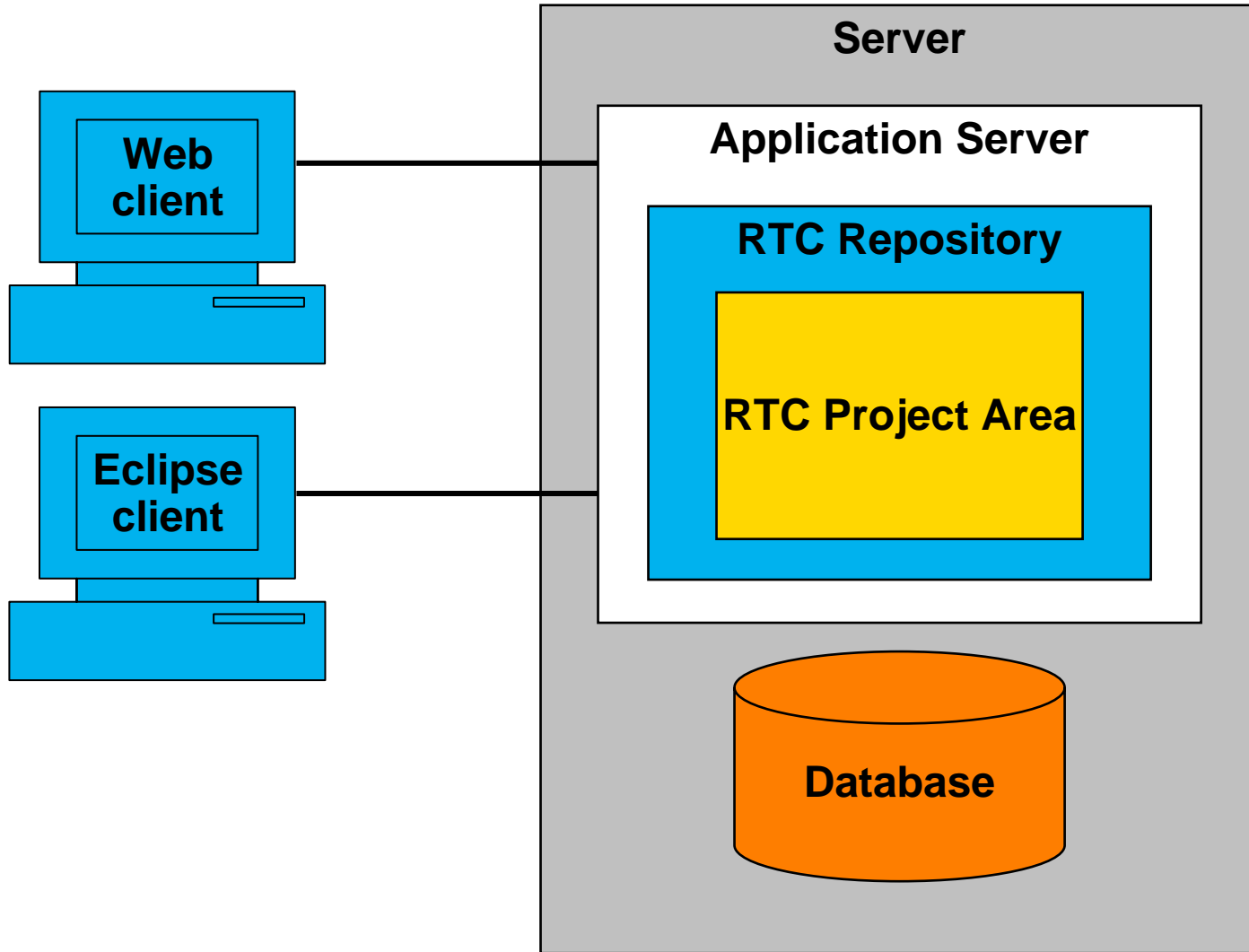


# Agenda

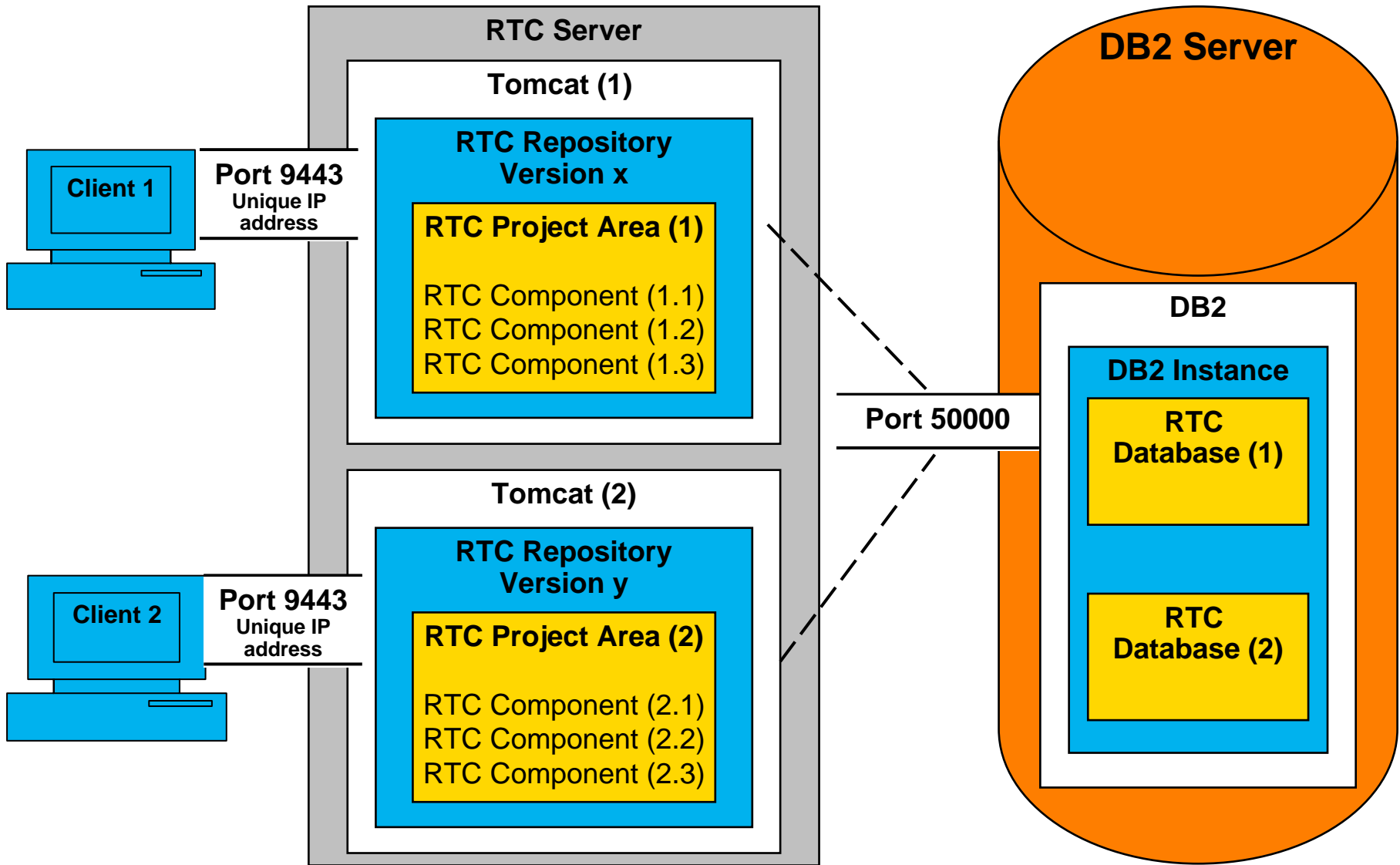
- Goal of presentation
- Centralized deployment of Rational Team Concert
- **A Simple RTC Configuration**
- Considerations for deploying Rational Team Concert
- Rational Software's RTC Deployment
- Benefits, Best Practices, and Lessons Learned



# A Simple RTC Configuration



# Initial Rational RTC Deployment Architecture



# Agenda

- Goal of presentation
- Centralized deployment of Rational Team Concert
- A Simple RTC Configuration
- **Considerations for deploying Rational Team Concert**
- Rational Software's RTC Deployment
- Benefits, Best Practices, and Lessons Learned

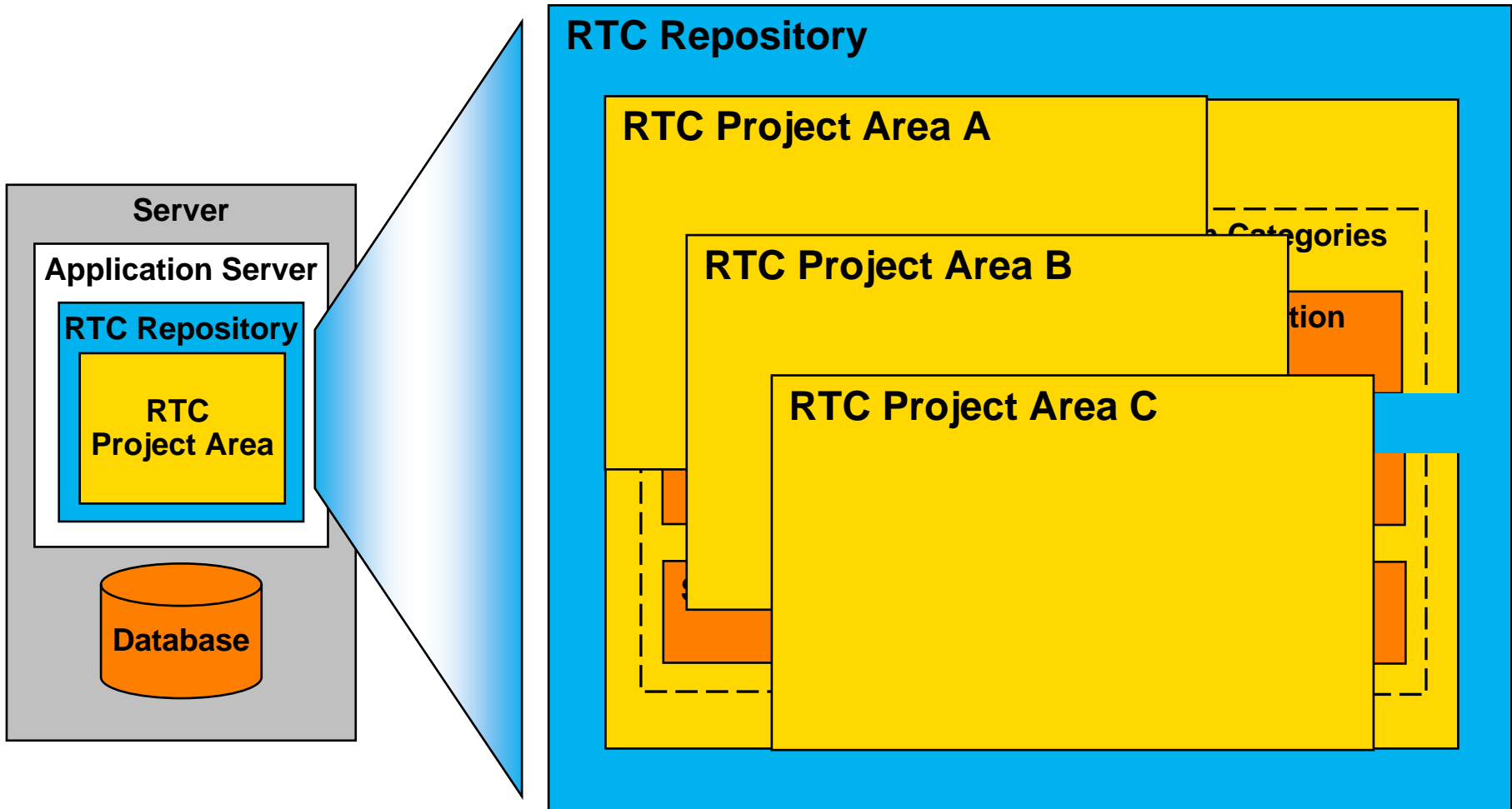


# What to take into consideration

- Before Deployment
  - ▶ Repositories vs Project Areas
  - ▶ Authentication/Authorization
  - ▶ Roles and Types of access
  - ▶ Infrastructure Considerations
  - ▶ Working with existing repositories
  - ▶ Connecting with ClearCase and ClearQuest
  - ▶ What edition of Rational Team Concert do you need?

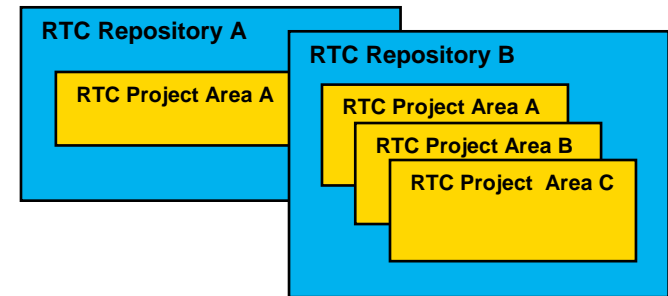


# Repositories vs.. Project Areas



# Repositories vs.. Project Areas

TIP → Determine whether to create single or multiple project areas per repository



## Single Project Area per repository

- ▶ Tighter controls
  - Keep source code confidential
  - Each repository has its own security domain
- ▶ Can't share workitems across repositories
- ▶ Multiple repositories with a single project use more server resources than multiple projects per repository
- ▶ Team controls its maintenance schedule
- ▶ Appropriate for a large team working on a single code base - up to 250 users per repository

## Multiple Project Areas per repository

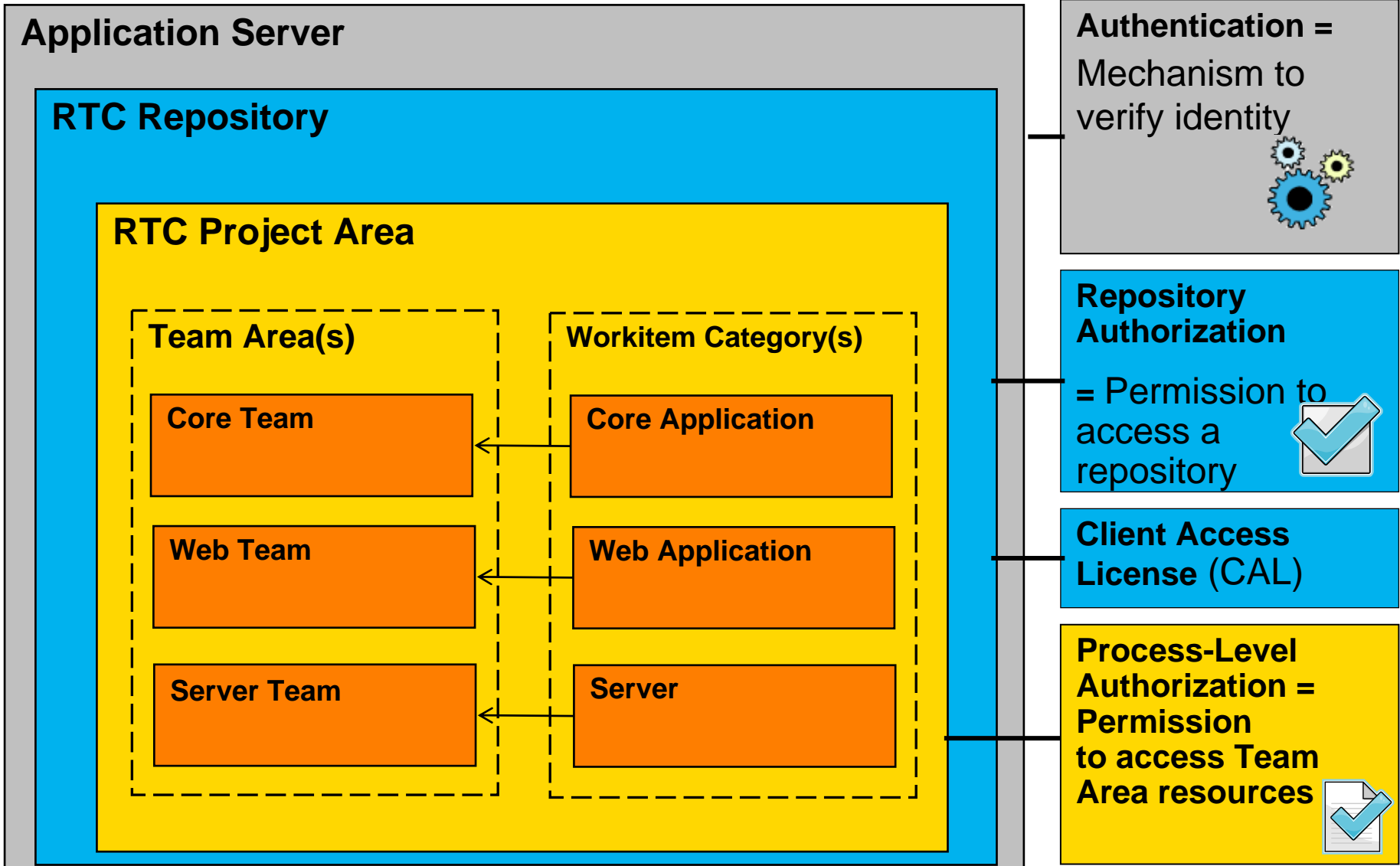
- ▶ Looser controls
  - Source code is viewable to all users of the repository (but write is restricted)
  - A user can attach to any Project area
- ▶ Can copy or move workitems between project areas within a repository
- ▶ Uses fewer server resources
- ▶ All teams must agree to RTC version & maintenance schedule
- ▶ Appropriate for multiple smaller teams working on related projects – limit is still 250 user per repository

**Rational has both single and multi-project repositories depending on needs of team**





# Authentication vs.. Authorization



# Authentication



- Tomcat
  - ▶ Supports authentication using either Tomcat Users or LDAP (but not concurrently)
  - ▶ Tomcat Users
    - Less secure
    - Stores account passwords in plain text on the server
  - ▶ Other forms of authentication require custom code
- WebSphere Application Server (WAS)
  - ▶ Enterprise level authentication
  - ▶ Provides federated repository capability
    - Can authenticate with more than one authority (i.e. LDAP(s), AD and local operating system accounts) concurrently
- Rational
  - ▶ Relies on Tomcat application server to provide authentication with IBM's Corporate LDAP directory



# Repository-level Authorization



- Repository Roles and Access → *Can I get into the server?*
    - Each role maps to one or more groups in the external registry
      - JazzAdmins
      - JazzDBAdmins
      - JazzUsers
      - JazzGuests
  - Licensing → *What can I do when I get there?*
    - Developer License
    - Contributor License
    - Licenses for system processes like Build and Connectors
- Rational
    - Uses internal Account Request database to request access to RTC projects
    - Combination of Developer and Contributor licenses per repository



## Process-Level Authorization



- ▶ Process Roles and Access → *Within my repository and license permissions, what can I do when I get there?*
  - More refined access control to components and functionality
  - Configured in process specifications per project

### THIS IS A QUIZ !!!

#### What controls these situations?

- You...
  - get a message that your user id is invalid
  - have permission to create new project areas
  - can close a work item for Core Applications but not for Web Applications
  - can create work items but cannot check in code
  - have permission to modify process specifications



# Infrastructure Considerations

- Supportability
  - ▶ What Hardware and Operating Systems are you comfortable supporting?
- Corporate Standards
- Existing Database Infrastructure
- Software Licenses
  - ▶ Operating Systems
  - ▶ Applications Servers
  - ▶ Database
- Capacity
  - ▶ Disk
  - ▶ Memory
  - ▶ CPU
- Network
  - ▶ IP Address space
  - ▶ Firewalls



# Rational Environment

- Hardware
  - ▶ 2 pSeries 9133-55A for both application and database server
  - ▶ 8 GB memory
    - 1 4-core 1.65GHz P5+ Processor Card
    - 8 146.8GB 10K RPM SCSI Drives (~1.2 TB/600GB mirrored)
- Software
  - ▶ Server – AIX 5.3 (not officially supported)
  - ▶ Application Server – Tomcat 5.5
  - ▶ Database – DB2 9.1
  - ▶ Clients – Windows & Linux
- Allocations for Users
  - ▶ Diskspace
    - Baseline of 1 GB of disk space per repository for application files
    - 250 MB of disk space per user
  - ▶ Memory
    - 1 GB per repository



# Working with existing repositories

- Other RTC Repositories

- ▶ **You Can**

- Copy and paste any process specifications or synch rules
- Copy source code
- Change application servers
- Migrate repositories from Derby to DB2 or Oracle
- Move or copy work items between project areas in a single repository

- ▶ **You Can't**

- Merge existing project areas into a single repository
- Move or copy work items between repositories
- Move a project area from one repository to another
- Split projects from a single repository into separate repositories
  - **Tip:** Make 2 (or more) copies and archive the non-needed

- External Repositories – You can

- ▶ Import from external sources

- Bugzilla
- CVS
- Subversion

- ▶ Share data with ClearCase and ClearQuest



## Connecting to ClearQuest and ClearCase

- ClearQuest Connector
  - ▶ Synch rules map data between RTC work items and ClearQuest records
  - ▶ Gateway server process exchanges data
- ClearCase Connector
  - ▶ Synchronizes streams between CC and RTC
- Both are RTC Install options available only in Standard Edition

### Rational

- Uses CQ Connector to synchronize work items with RATLC defect records
- Some constraints with version 1

▶ An RTC Repository may only connect to one ClearQuest database

▶ Limited support for CQ Multisite.

- Gateway can only communicate with one CQ replica CQC will only update CQ records mastered at that replica.
- **Mitigation:** Maintain mastership of all synched records at a single replica – access via web client





# What edition do you need?

## Express-C

➤ **Consultants and students**

- In-context collaboration
- Rapid deployment
- Open source middleware

## Express

➤ **Small and medium businesses**

- Transparent development
- Out-of-the-box processes and process enactment
- Commercial middleware

## Standard

➤ **Mid-sized companies and smaller teams in large companies**

- Customizable process
- Real-time project health
- Enterprise scalability and extensibility

Maximum users per server	10	50	250
Middleware	Tomcat, Derby	DB2 Express, DB2, Oracle, Tomcat or WAS	DB2 Express, DB2, Oracle, Tomcat or WAS
Authentication	✓	✓	✓
Server-level user permissions	✓	✓	✓
Off-the-shelf process	✓	✓	✓
Subversion Integration	✓	✓	✓
Process customization			✓
User Import and Synchronization			✓
Proxy Support – when available			✓
Customizable Work Items			✓
Role-based Process Permissions			✓
Customizable Reporting and Alerts			✓
User Customizable Dashboards			✓
ClearCase, ClearQuest Integration			✓

**Rational uses this version**



# Things that come into play after deployment

- ▶ Monitoring
- ▶ Scalability and Expandability
- ▶ Disaster Recovery
  - Backup/Restore
  - Relocation of services
- ▶ Ongoing user management



# Monitoring

- Database server
  - ▶ Database table size
  - ▶ Disk space
- Application Server
  - ▶ CPU usage
  - ▶ Memory consumption
  - ▶ Notices in application server log file

→ Rational uses Tivoli Monitoring and custom scripts



# Scalability

- Scaling up
  - ▶ Simplest: Single Server with Application and Database on one server
  - ▶ Scaling Up - Step 1: Split up Application Server from Database Server
  - ▶ Scaling Up - Step 2: Add more Application Servers (up to 250 users each)
- Considerations for scalability
  - ▶ Hardware
    - Make sure your machine has enough memory and disk to handle anticipated growth
    - To prevent bottlenecks, don't skimp on memory or disk
  - ▶ Application server
    - Add more threads to Tomcat to allow more simultaneous connections
  - ▶ Database
    - Don't start with Derby if you think you're going to grow quickly



# Disaster Recovery

## ■ Considerations

- ▶ Database backups or resiliency
- ▶ Backing up RTC server code and configurations
  - teamserver.properties
  - web.xml
  - server.xml (for Tomcat)
  - Workitem index (directory)
  - Profile backups (for WAS)
- ▶ No need to back up the installed RTC packages
  - If you can always access RTC install and database software

## ■ RTC Recovery process

- ▶ Restore the database; roll forward using archive logs
- ▶ Deploy RTC to your application server
- ▶ Replace configuration files and reapply licensing files

## ■ Rational

- ▶ Does nightly database online backups, then uses TSM to push to tape and offsite storage. We retain the 3 most recent copies on disk.



# Agenda

- Goal of presentation
- Centralized deployment of Rational Team Concert
- A Simple RTC Configuration
- Considerations for deploying Rational Team Concert
- **Rational Software's RTC Deployment**
- Benefits, Best Practices, and Lessons Learned

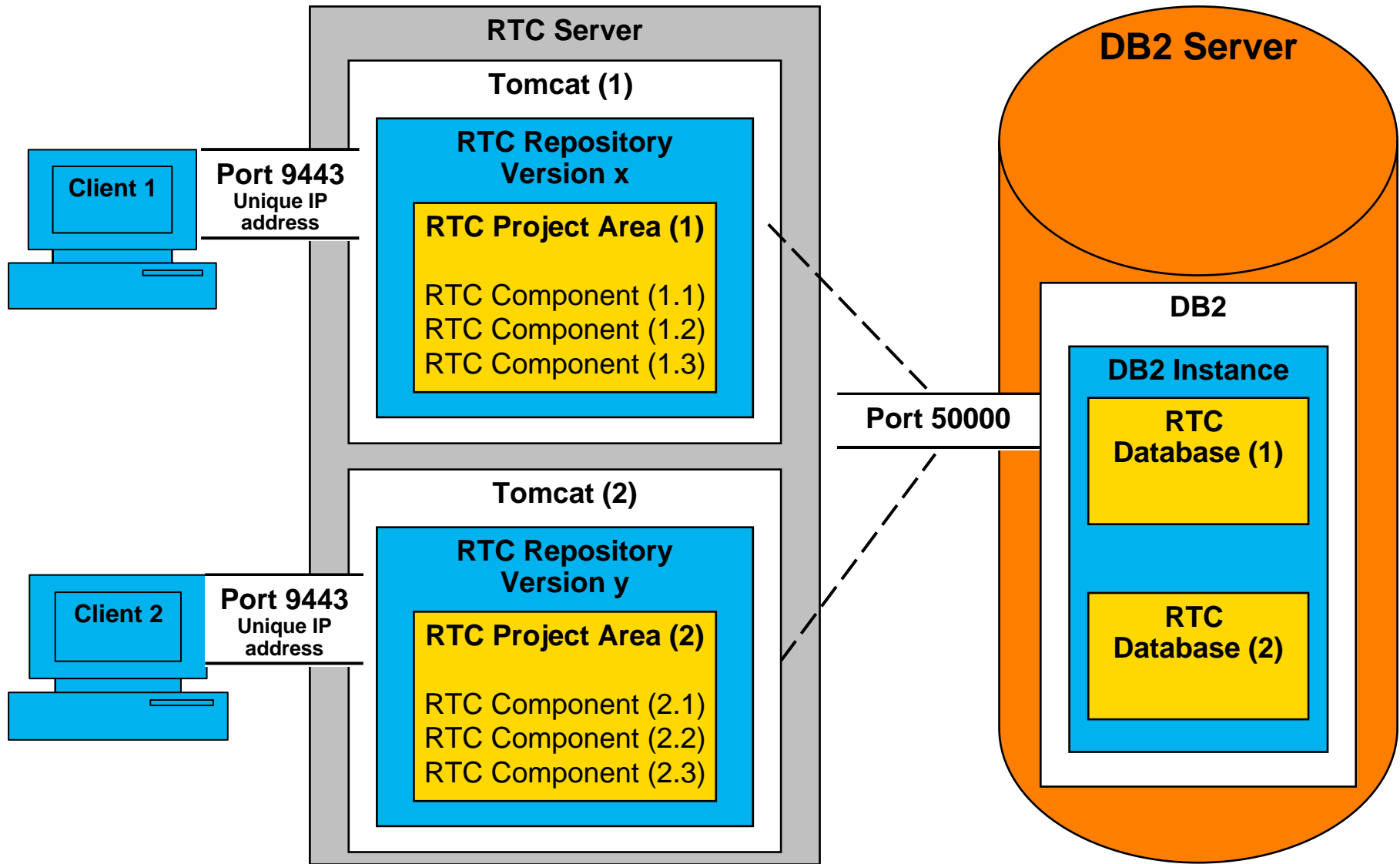


# Rational Software's RTC Deployment

- Deployment architecture
- Process for setting up new teams
- Setup steps and ongoing maintenance

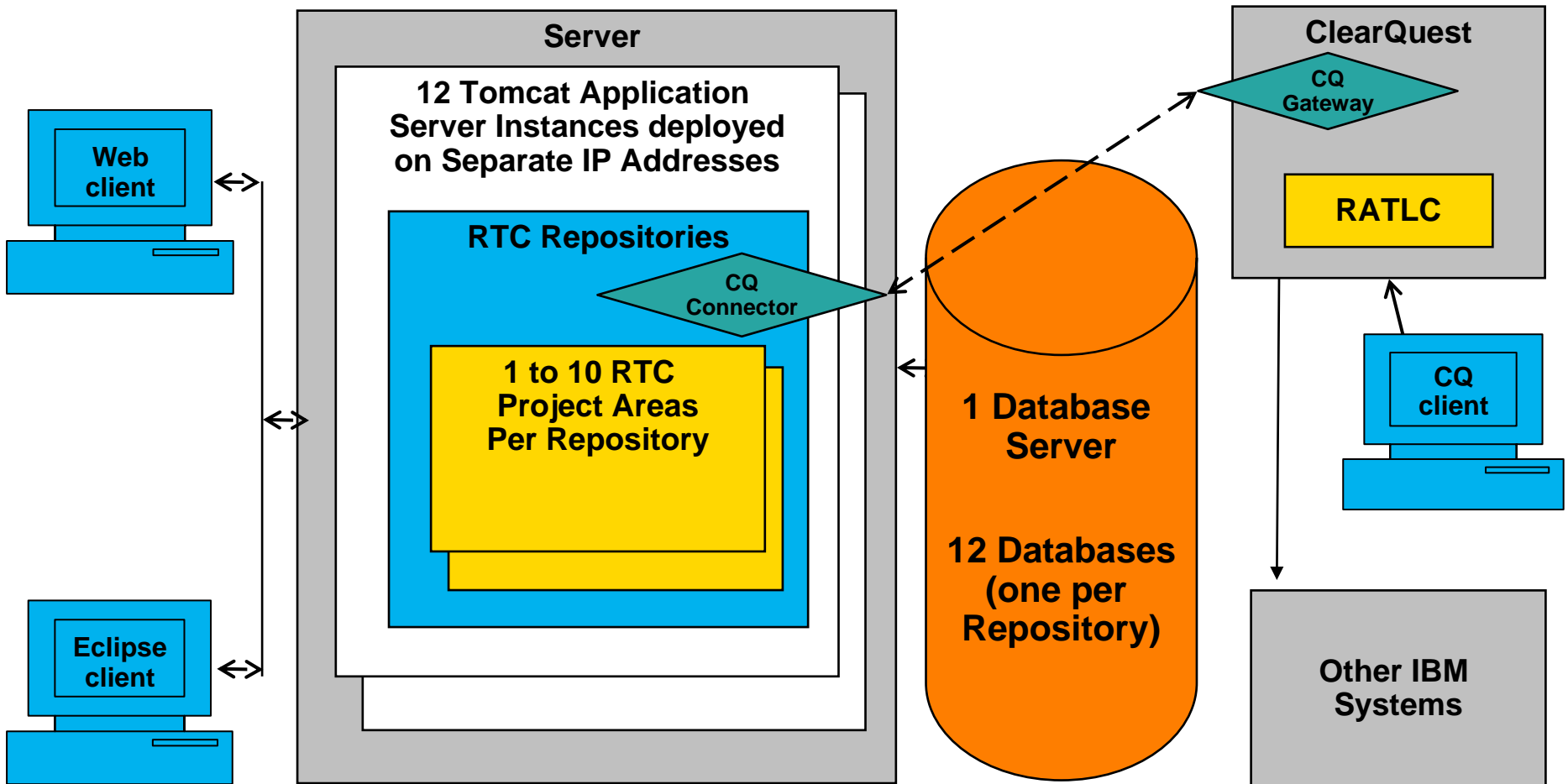


# Initial Rational RTC Deployment Architecture





# Current RTC Deployment Architecture



## Our process for setting up new teams to use RTC

- Questionnaire regarding goals, size of team, etc (see Appendix A)
  - ▶ Helps RES determine size of deployment and planned usage
  - ▶ Requires approval of senior management – used to provide visibility to adoption rate
- On-boarding process
  - ▶ Slide presentation and meeting with new team contacts to set expectations
  - ▶ Answer questions and understand roles and responsibilities
- Service level agreement for hours and support (see Appendix B)
- Definition of roles and responsibilities (see Appendix C)
- Sandbox environment for first endeavors
- Communications - [Wiki with schedules](#), contact and support info, etc.,
- Ongoing requests from users
  - ▶ RTC workitem in RES's project area!



# Setup steps and ongoing maintenance times

## Initial Setup (once per app server)

- Configure app server
- Configure database
- Set up LDAP or other registry

## Repository & Project Area Setup

- Create RTC user groups/roles
- Ops adds users to repository and assigns licenses
- Project owner adds users to teams and project roles (team lead, contributor, etc.)

## Ongoing Maintenance Time Estimates

Adding users to new server	A few minutes per user or can be scripted
Setting up a production repository on an existing server	2-3 hours
Setting up a project area in an existing repository	< 10 minutes
Migrating to a new release	Depends on size of project repository
Migrating Jazz.net	~ 14 hours for one 40-Gb database
Migrating smaller repositories	~ 14 hours for 5 4-Gb databases



# Agenda

- Goal of presentation
- Centralized deployment of Rational Team Concert
- A Simple RTC Configuration
- Considerations for deploying Rational Team Concert
- Rational Software's RTC Deployment
- **Benefits, Best Practices, and Lessons Learned**



## Operational Benefits of using RTC

- Easy to set up
- Easy to scale
- Low total cost of ownership (TCO) and administrative overhead
- Easy to back up and recover
- Good for distributed teams
- Faster setup, lighter-weight than ClearCase
- Connects to existing ClearQuest and ClearCase repositories



## Best Practices and Lessons Learned

- Set up a sandbox to allow people to “play” with their process and customizations prior to actual deployment
- No need to put each repository into its own physical or virtual server
- Push teams to really consider collaboration needs
  - ▶ Do they need to share workitems
  - ▶ Not easy to transplant Project areas
  - ▶ No RTC/RTC Connector between repositories
  - ▶ No read restrictions on access between projects in a single repository
  - ▶ Having single RTC version makes life easier for centralized build teams
  - ▶ Coordinating upgrades for multiple projects within a single repository requires high level negotiation skills!
- You can scale up from a smaller installation, but if you know you’ll be growing quickly, size your hardware appropriately
- CQ Connector
  - ▶ Start small – use the sample database to learn how to set up the synch rules
  - ▶ v1 CQ Connector has limited MultiSite support





# QUESTIONS





# THANK YOU

## Learn more at:

- [IBM Rational software](#)
- [IBM Rational Software Delivery Platform](#)
- [Process and portfolio management](#)
- [Change and release management](#)
- [Quality management](#)
- [Architecture management](#)
- [Rational trial downloads](#)
- [Leading Innovation Web site](#)
- [developerWorks Rational](#)
- [IBM Rational TV](#)
- [IBM Rational Business Partners](#)

© Copyright IBM Corporation 2008. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, the on-demand business logo, Rational, the Rational logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.





# References / Suggested Reading

- [jazz.net](http://jazz.net)
- Team Concert Help Menu
  - ▶ Installation and upgrading
    - Simple installation worksheet
    - Server Installation Planning – includes list of supported platforms
    - Client Installation Planning – includes list of supported platforms



# RTC License Model



\*Feature provided by Build Forge

\*\* Features in Standard Edition



Business value

- Team First
- Transparency
- Collaboration
- Integration

## Collaborative Development

- Focus on development team productivity
  - Enables distributed development
- Flexible support for development processes
- Manage Change in the Application Lifecycle



Developer

- 100s users
- Deep tool/programming language knowledge
- Talks technology language
- Distributed teams

## Source Code Management

- Workspaces
- Snapshots
- Baselines
- Change Sets
- Suspend Changes
- Component
- Streams
- SVN Bridge
- **ClearCase Connector\*\***

## Work Items Management

- Iteration planning
- Attachments
- Audit trails
- Customize workflow\*\*
- **ClearQuest Connector\*\***

## Build Management

- Define Builds
- Continuous integration
- Remote Build Server
- Bridges
- *Build Forge integration\**

## Team Health

- Transparent development
- Team Central
- RSS/Atom Feeds
- Instant Messaging
- Process Enforcement and Customization
- **Dashboards\*\***
- **Reports\*\***
- **Role-based Permissions\*\***



Contributor

- Web based UI
- Knows application domain
- Talks business language

## Team Collaboration

- View Iteration plans and validate deliverable time lines
- Create, Update, Query and view disposition of development tasks (work items)
  - **Overview of the resource utilization\*\***
  - **View personalized Dashboards\*\***

IBM & Partner Ecosystem



# Appendix A: RTC Hosting Request Form

1.	What is your project name?
2.	Please provide a brief project description
3.	Requested start date for hosting
4.	Who will be signing the service agreement?
5.	What is your Group, Division and Department?
6.	Who is your financial analyst?
7.	Who is the primary point of contact for this project?
8.	Who is the project owner or manager?
9.	Who will be your RTC administrator for this project?
10.	What kind of artifacts will you be storing? ( Work Items, Builds, Plans, Requirements, Streams)
11.	How many developer & contributor licenses do you anticipate needing for this project?
12.	If you are moving from an existing RTC Server, please provide a link.
13.	Estimated initial disk space requirements in GB (for source code & attachments.)
14.	Estimated long-term space requirements in GB:
15.	Is your project currently shared with any other group? If so, what groups?
16.	Do you have additional requirements, such as plugins or connectors? If so, please describe.
17.	Do you need consulting assistance for bringing existing artifacts into the RTC server?
18.	What is your project or team wiki/website?
19.	Comments, and other information.



## Appendix B – Support Coverage

- Support coverage 24x7 for these outage situations:
  - ▶ Hardware, OS, RTC application, RTC database, and Network outages. Applications or major features are nonresponsive or hung.
  - ▶ Major features are defined as: (workitems, client connections, web UI, builds)
  - ▶ Backup and restore (Catastrophic recovery)
- Support coverage 9x5 (ET)

The RTC Operations Support team is located in the Eastern time zone (U.S.). '9x5' Coverage means 'standard business hours' EST. (National and site holidays are excluded.) RES supports these service situations:

  - ▶ Network Performance slowness  
If RTC itself is running slowly, RES will investigate and do what we can, within the networking environment.
  - ▶ OS and Application support : APARS and upgrades (happen during outage periods, whenever possible)
  - ▶ Scaling/Capacity issues regarding: CPU, Memory, Disk space, Application usage



## Appendix C: Roles & Responsibilities - RES

- Post connection point information (URI) for the various RTC Projects.
- Provide version information of RTC Server/Clients.
- Migrate data as part of upgrade process.
- Coordinate the RTC upgrade schedule.
- Provide debug hooks in RTC to create core/heap files, (for RTC Dev team debug).  
Tivoli monitoring, which can be made available for RTC Dev team debug.  
Nightly backups.
- Post/administer outage schedules for the development teams  
(with some level of flexibility, if needed - on a case by case basis)
- Central point of communications (wiki) for:
  - RTC Project Admins
  - RTC Project Owners
  - RTC Project Managers
  - Outages, upgrades, etc.
- Compliance
  - ▶ Health Checks, APARS, Scans, etc



## Appendix C – Roles and Responsibilities - Users

- **RTC User Responsibilities** (links require a jazz.net ID)
  - ▶ Users sign up to use RTC via a Rational account management tool
  - ▶ Operations team imports user from corporate LDAP directory, adds user to repository and assigns license
  - ▶ RTC Project Admin adds user to their team area
  - RTC user training is user's responsibility, start here: [Learn About RTC](#)
  - ▶ Install and upgrade RTC Clients

- **RTC Administrator Responsibilities**

An admin is responsible for configuring and maintaining RTC development processes including:

- ▶ [Project Areas and Process](#)
- ▶ [Work Items](#)
- ▶ [Source Control](#)
- ▶ Admin must train on their own. Here are links to training materials:
  - [Builds](#)
  - [Iteration Planning](#)
  - [Reports](#)
  - [Learn About RTC](#)
  - [Getting Started With RTC](#)
  - [Demo videos](#)
  - [Tech Notes](#)
  - [RTC Platform Technical Overview](#)
  - [Team Server Setup Guide](#)
- ▶ Aside from RTC-specific responsibilities, admins must add and manage RTC users through [RTC Admin UI](#).

