Multi-User Development of an ArchestrA Galaxy: Best Practices

Introduction

Simultaneous multi-user development of a single ArchestrA galaxy is often needed for a project, particularly when the project is large. The multi-user environment can be set up with multiple concurrent IDE connections from either standalone computers within the galaxy or terminal client sessions (see the picture below).

However, the advantages of this development environment (time-saving being the most obvious one), can be offset by some commonly identified issues that arise from this configuration.
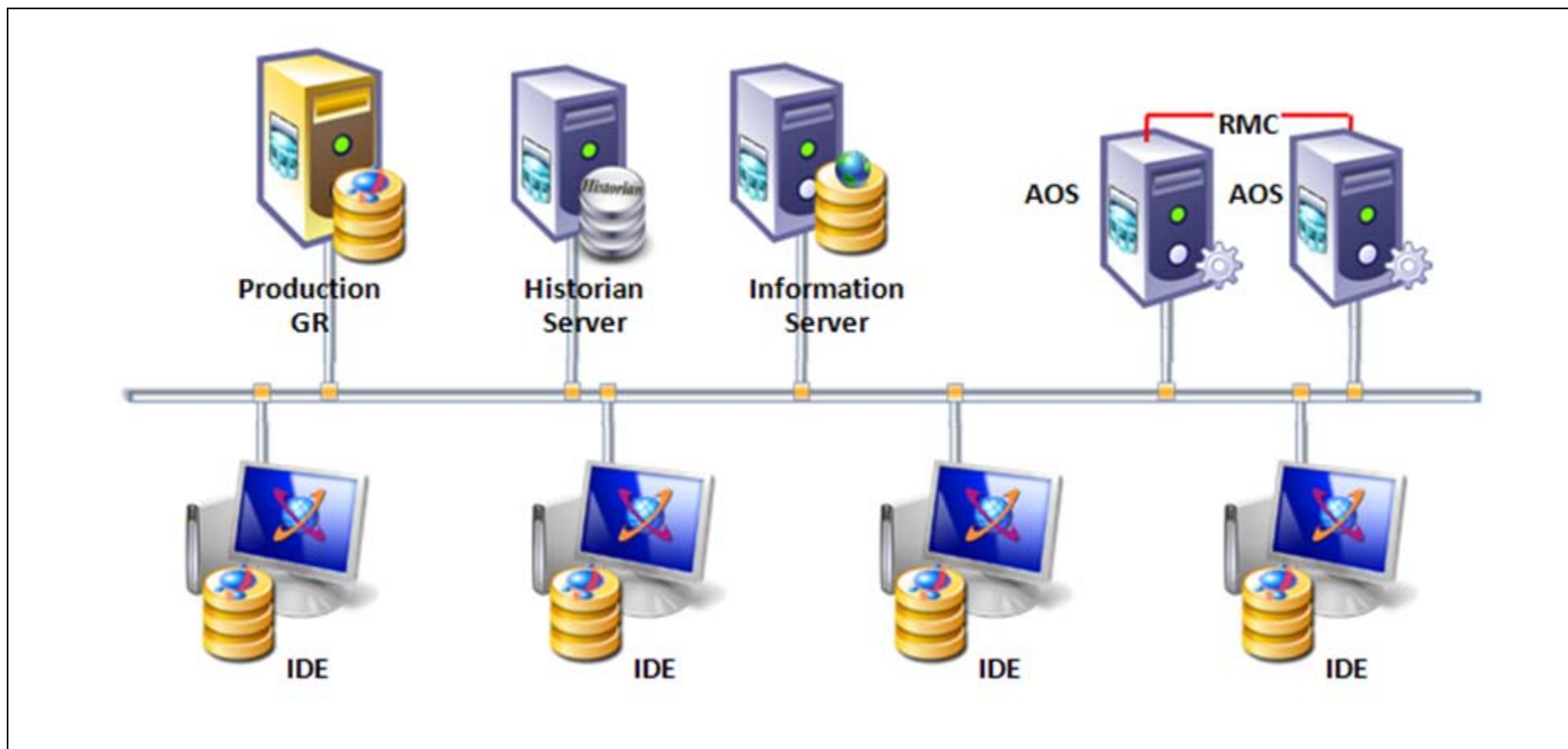
Some common issues are listed below:

- Long check-in and check-out time. It even times out after a long wait.

- IDE not responding, aaGR service hanging after check-in or check-out is submitted, sometimes, IDE even crashes.

- Development of objects interfering with the system deployed and in production.

- ArchestrA Security update that disrupts other users' IDE connections.

Application Versions

- Wonderware Application Server 3.0, 3.1

Solution: Distributed Multi-User Development

To get around these issues with multi-user development, the following practices are highly recommended.

1. First, create a separate galaxy in addition to the production galaxy on the same box. For this Tech Note, it's called **Template Galaxy**. The Template Galaxy is used as a Master Template storage location. The Template Galaxy is used to

- Maintain the current master templates for all objects.

- Distribute master templates to Local GR for development (via .aaPkg export).

- Control master template updates from local developers (via .aaPkg import).

- Maintain ownership relations between the developers and master templates (via enabled ArchestrA security---Galaxy, OS User Based or OS Group Based).

    Figure 2 (below) shows levels of hierarchy as an example. The levels dictate the ownership of the templates within the team/organization.
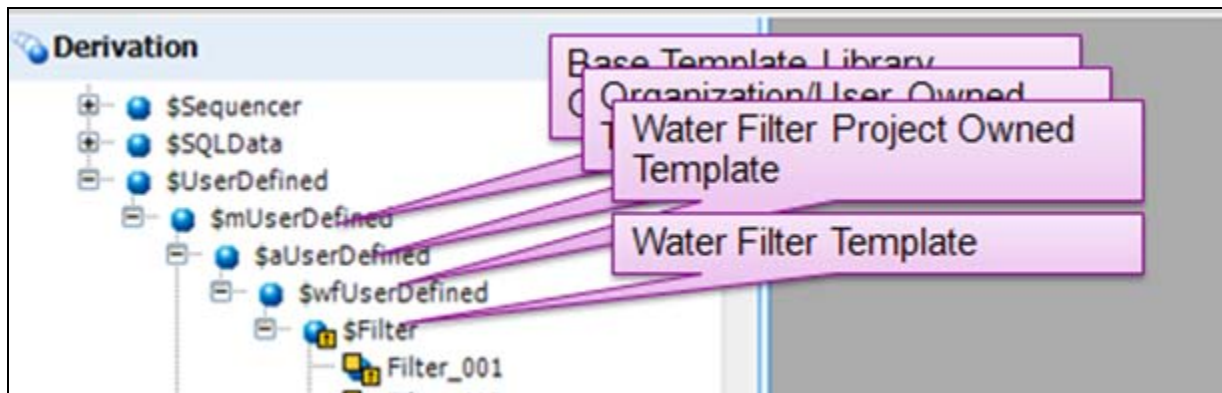

FIGURE 2: TEMPLATE OWNERSHIP ENABLED BY ARCHESTRA SECURITY

2. Install a temporary GR called **Local GR** on each IDE development node. The Local GR ensures that each developer can work

independently without interference from other developers' check-in/check-out, security updates, and deployment actions.

The Local GR can also serve as a testing bed in addition to development.

Note: The Local GR should not require additional platform count license because it is first, a temporary installation; secondly, it is not for runtime use.
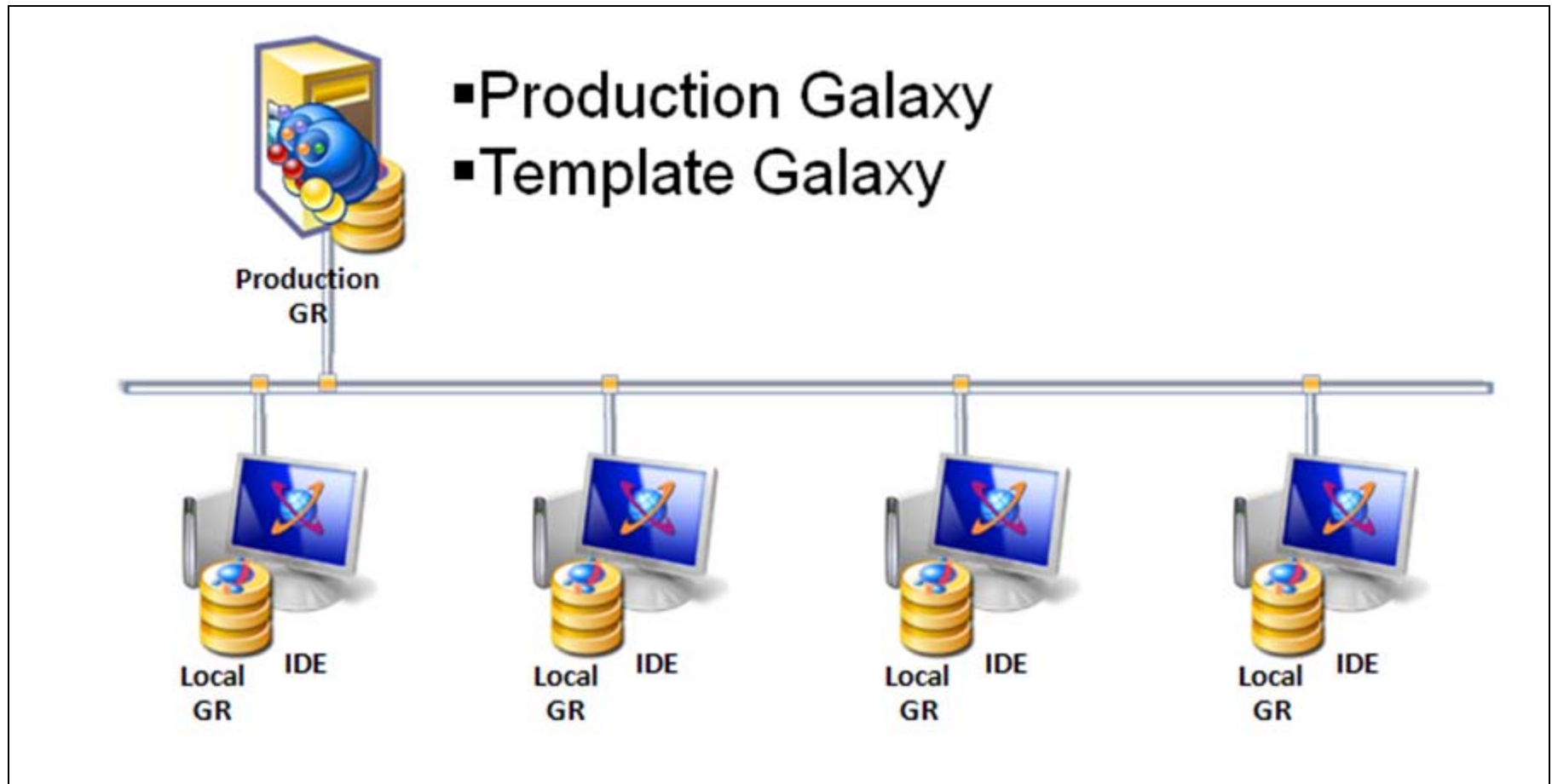

FIGURE 3: LOCAL GRS ON MULTIPLE DEVELOPMENT NODES

Development Process Steps

1. Login to the Template Galaxy.

2. Select and Check Out template(s) you need for development.

3. Export template(s) as package (.aaPkg) from Template Galaxy.

4. Login to your Local GR.

5. Import the package into your Local GR.

6. Edit/Deploy/Test from your Local GR.

7. Export as package from your Local GR.

8. Login to the Template Galaxy.

9. Undo the Checkout.

10. Import the package from Local GR to the Template Galaxy.

11. Checkout, then Checkin and add comments in order to document your development efforts.

Details of Production Galaxy Updates

- Back up the existing Production Galaxy as necessary.

- Transfer Templates to the Production Galaxy when the project is ready for an updates.

- Create new object instances in the Production Galaxy as necessary.

Benefits of Distributed Multi-User Development Strategy

- Several engineers can develop simultaneously.

- They can always rollback changes from unwanted experimentation.

- The Production Galaxy is isolated, so there is no disturbance/disruption on the Production Galaxy from development efforts.

- You can manage and control your update release to the Production Galaxy.

- No check-in/check-out, and/or long wait and timeout.

- No IDE non-response and the aaGR service doesn't hang.

- No IDE connection interruption from security updates.

C. He

For technical support questions, send an e-mail to **support@wonderware.com**.

▲ **Back to top**