

19^c ORACLE[®]
Database

Oracle Multitenant with Oracle Database 19c

ORACLE WHITE PAPER | MARCH 2019





Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



ORACLE®



Table of Contents

Disclaimer	1
Introduction	2
Advantages of Multitenant Architecture	3
Multitenant's Comprehensive Isolation Architecture	4
Multitenant Use Cases	6
Key Benefits of Multitenant	8
Summary	9



Introduction

Oracle Database 19c rounds out the Database 12c family and with this release Multitenant is correspondingly rounded out, with fully fledged functionality. With 2019 upon us and with it the release of 19c, it's gratifying to look back on the evolution of Multitenant, from when it was first announced with Oracle Database 12c back in 2012 to what is now a mature, stable and well-established product.

The "c" in "19c" is for Cloud, of course, and Multitenant is Oracle's Cloud Database Architecture. The phrase that sums up the promise of the Cloud is "isolation and agility with economies of scale". Our goal all along has been to deliver on that promise, consistent with this phrase, as is explained in this White Paper.

Advantages of Multitenant Architecture

Let's review Oracle Multitenant – the container database architecture – the first database designed for the Cloud. We have these self-contained “pluggable databases” or PDBs for each application or tenant. From the point of view of an application, the PDB is the database, in which applications run unchanged. PDBs can be very rapidly provisioned and a pluggable database is a portable database, which makes it very easy to move around, perhaps for load balancing or migration to the Cloud.

Many PDBs can be plugged into a single Multitenant Container Database or CDB. From the point of view of a DBA, the CDB is the database. Common operations are performed at the level of the CDB enabling the DBA to manage many as one for operations such as upgrade, configuration of high availability, taking backups; but we retain granular control when appropriate. This ability to manage many as one enables tremendous gains in operational efficiency.

Enormous gains in technical efficiency are enabled by a shared technical infrastructure. There's a single set of background processes and a single, global memory area – the SGA – shared by all the PDBs. The result is that with this architecture we can consolidate more applications per server.

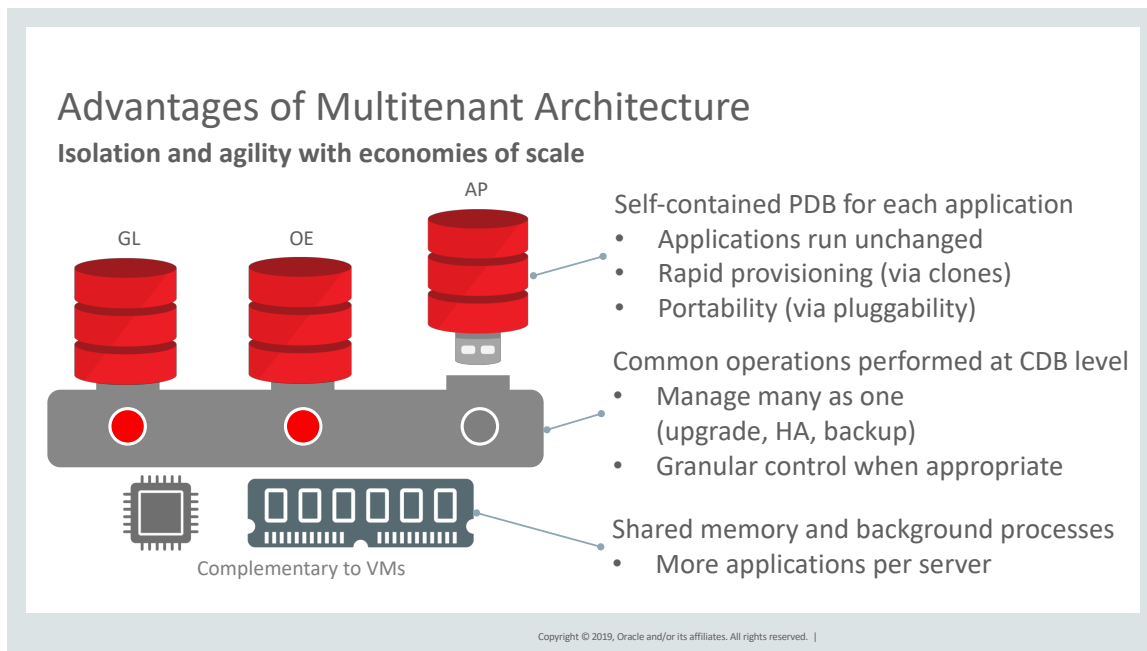


Figure 1. Advantages of Multitenant Architecture

These huge gains in operational and technical efficiency deliver the economies of scale promised by the Cloud. Let's review the other critical components, starting with isolation.

Multitenant's Comprehensive Isolation Architecture

Consistent with our security philosophy of “defense in depth”, Multitenant has a comprehensive isolation architecture. There are four major categories to this, with several important features in each category. These are summarized in Figure 2.

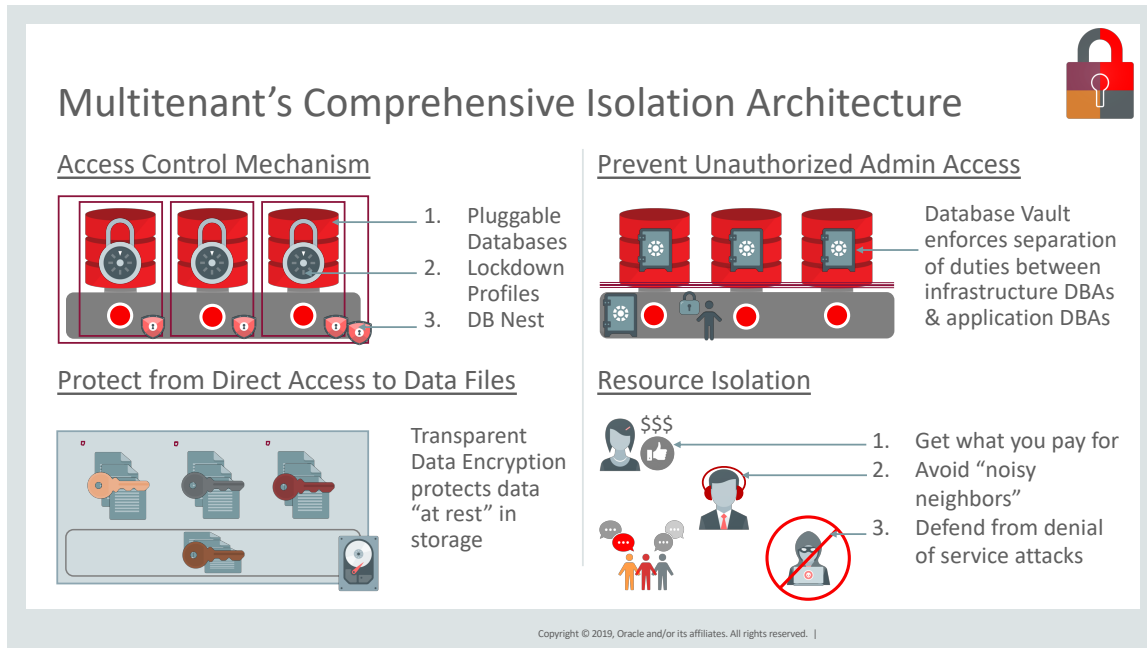



Figure 2. Multitenant's Comprehensive Isolation Architecture

1. From a security perspective, Multitenant is an access control mechanism.
 - a. The intrinsic architecture features self-contained Pluggable Databases for each tenant or application.
 - b. To this we add Lockdown Profiles
 - c. And, new in 19c, PDB Nest.

With each successive release, as we have added more and more security features such as lockdown profiles and PDB Nest, it has become possible to enable more and more functionality without compromising security.

2. In some use cases, such as Software as a Service (SaaS), it is important for infrastructure DBAs to be unable to see data in individual PDBs. Database Vault enables separation of duties between infrastructure DBAs and application DBAs. This allows us to prevent unauthorized access by *authenticated* (common) users to data local to the PDB. We might think of this in terms of “locking the front door”.
3. It is also necessary to protect data “at rest” in storage. Transparent Data Encryption provides this protection. Each PDB has a separate encryption key. We might think of this in terms of “locking the back door”.
4. Resource Management is the fourth aspect of this comprehensive isolation architecture.

- 
- a. A basic, common requirement is to have a means to “get what you pay for”.
 - b. Next, it’s important to avoid interference from “noisy neighbors”.
 - c. An important application of Resource Manager is to avoid denial of service attacks.

To address the requirements of a variety of use cases, configurable isolation is an important capability. Multitenant’s comprehensive isolation architecture enables the appropriate isolation characteristics to be configured in every case.

Multitenant Use Cases

Multitenant is now the standard architecture for Oracle Database and it follows that Multitenant's container database architecture is suitable for the full range of Database use cases. Figure 3 illustrates three prominent use cases, which are described in a little more detail below.

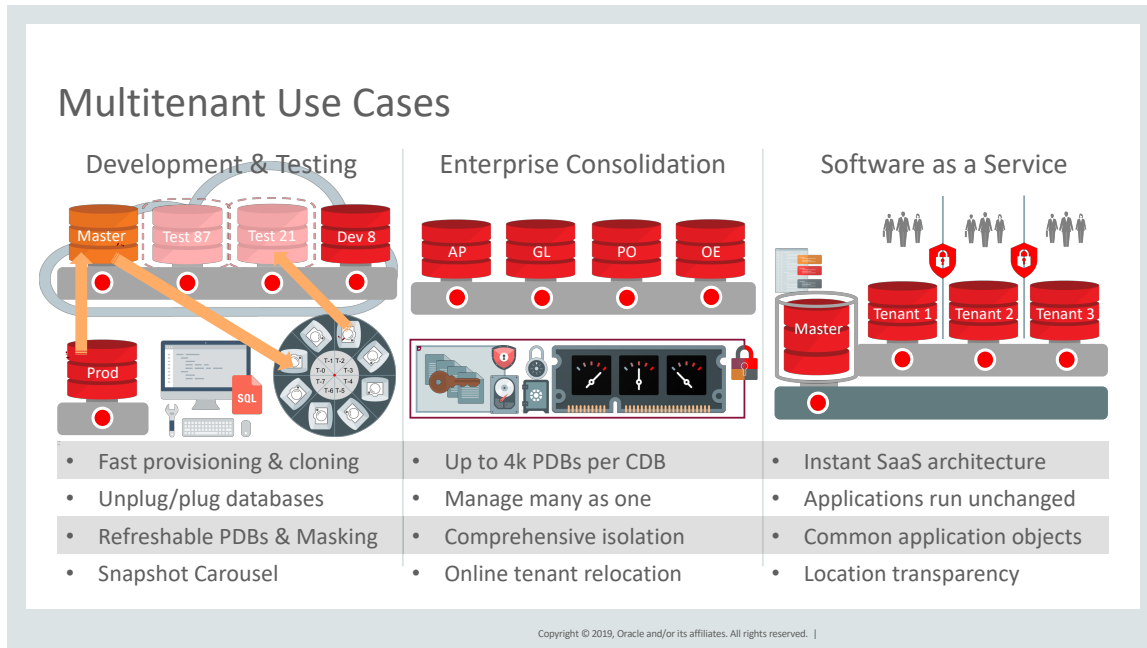



Figure 3. Three Prominent Use Cases for Multitenant

Multitenant is particularly well suited to development and testing.

- The low-load characteristics of development make PDBs dramatically more efficient than a VM-based model in these environments.
- The rapid provisioning of hot clones and refreshable clones enables very agile project teams.
- Debugging elusive time-specific problems becomes trivial with Snapshot Carousel.
- Storage costs and provisioning times are reduced by orders of magnitude with snapshot clones.

Enterprise Consolidation is a classic use case for Multitenant.

- With support for up to 4,096 (4k) PDBs per CDB in Oracle Exadata and on Oracle Cloud, and 252 PDBs per CDB on other platforms, very large-scale consolidation is possible.
- Tremendous operating efficiencies can be realized because many PDBs can be managed as one. For example, by applying a critical security patch to the CDB, all the PDBs are protected in a single operation.
- Despite being consolidated in a single CDB, each PDB is isolated, from “noisy neighbors” as well as from a security perspective.
- Online tenant relocation enables granular control without compromising availability.



Software as a Service is perhaps the use case after which the product is named.

- Multitenant delivers an instant SaaS architecture, with no application changes required.
- Location transparency enables agile load balancing without impacting business-wide administration.
- Common objects are defined and maintained in a single application master, enabling trivial application maintenance.



Key Benefits of Multitenant

There is a four-fold value proposition of Multitenant:

1. Minimize Capital Expenses
 - More Applications per server
2. Minimize Operating Expenses
 - Manage many as one
 - Standardize procedures and services
 - Enable self-service provisioning
3. Maximize Agility
 - Rapid provisioning
 - Portability through pluggability
 - Scalability with RAC
4. And it's easy!
 - Easy to adopt, because no application changes are required
 - Easy to use, because SQL (and svrctl) are the interfaces.

With traditional approaches to consolidation, there's typically been a tradeoff between economies of scale and agility. With Oracle multitenant, you can have your cake and eat it.



Summary

Multitenant is the fundamental architecture for Oracle Autonomous Database. The acid test of the success of this architecture is that we have been able to build this highly advanced database cloud service with Multitenant at the core. We pass the test with flying colors!



Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

-  blogs.oracle.com/multitenant
-  facebook.com/oracle
-  twitter.com/OraclePDB
-  oracle.com/goto/multitenant

Integrated Cloud Applications & Platform Services

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. This document is provided *for* information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0116

Oracle Multitenant with Oracle Database 19c
March 2019
Author: Patrick Wheeler, Senior Director, Product Management, Oracle Database