CA ERwin Data Modeler's Role in the Relational Cloud

Nuccio Piscopo

ca technologies

ERwin

www.ERwin.com

Table of Contents

Abstract

The Database as a Service (DaaS) model is growing its presence across the virtual globe. Enterprises are asking whether DaaS is mature enough to meet their low-cost and high-security data management requirements using a cloud paradigm. Companies are deploying virtualized applications and, as a consequence, they are asking for Relational Cloud as well. Dynamic scalability, privacy, performance and heterogeneous environments or interfaces to business intelligence products ask for a class of solutions to satisfy the cloud paradigm as a whole. Data modeling supports DaaS in that it provides a clear mapping of the deployed data structures, storage and data topology as well as the core business concepts. In this way, CA ERwin Data Modeler enables organizations to collect and serve data models from and to any web data source and data management system in the cloud.

Introduction

DaaS involves the process of moving operational provisioning, configuration, scaling, performance, backup, privacy and access control from the database users to the service operator, offering lower overall costs to companies. Looking at Relational Cloud this means DaaS has to address the following challenges:

1. Efficient multi-tenancy
2. Dynamic scalability
3. Database privacy

CA ERwin Data Modeler works to align company models and databases used as a service by providing the following:

1. Ability to create models in heterogeneous database environments
2. Ability to interface with business intelligence products
3. Support of diverse ud architectures (shared-nothing, shared-disk, hybrid)
4. Functioning irrespective of replication over large geographic distance or database parallelization or time they are executed

Customers asking for Relational Cloud request large-scale, multi-node DaaS solutions by presenting and preserving the design principles and implementation status.

As a consequence, this paper introduces the ERwin Relational Cloud and describes the concept of MaaS or Modeling-as-a-Service. CA ERwin Data Modeler is one of the main actors in the Relational Cloud and supplies "a priori" database design configuration, model security, privacy and database performance both to a single organization with many individuals databases (private cloud) and to a public service that allows database provisioning and administration to be outsourced to a third party provider (public cloud).

DaaS requirements through CA ERwin Data Modeler

Looking at the Relational Cloud, we can complete the service model shown in by using the following matrix to associate DaaS requirements to CA ERwin Data Modeler functions:

Customer Requirements	CA ERwin Data Modeler Functions
Near-zero API configuration and administration	Models contain all database properties including scripts and stored procedures
High-performance	Verify multi-database structure performance and measure query timing
High availability	Assure structure validation in terms of model accessibility
Fast updates at low cost	Database support for forward/reverse engineering is continually updated as DBMS vendors add new features.
Multi-database partitioning	Support for partitioning models to database partitions (create, update, move-from-to) and managing multiple databases with a single model
Migration	Database migration can be managed by comparing model to model using the multi-database partition tool

Provider Requirements	
Meet customer SLA agreement	Database model performance covers a large part of the SLA agreement.
Limit hardware and power costs	Platform-independent multi version DB support limits costs needed for separate installations.
Limit Administration costs	Mitigate database administration through model administration. Rather than relying on in-house skills to manage multiple database platforms, the modeling tool creates database-specific syntax.
Cover continuity	Models cover continuity. Models are versioned and model change management is applied.
Integrity	Models (i.e. database structures) maintain integrity. Providers having a model change life cycle can assure integrity and protected integration, partitioning, and migration.
Public Cloud Requirements	
Pricing scheme: cheap, predictable and proportional to actual usage (elasticity)	Database models cover this need. Except when usage depends upon network connection, multi-database usage can be estimated in terms of detailed model properties.
Risk mitigation: security and privacy	Models are the only way to enable data protection within the database. The model should be shared and updated to be synchronized. It can also be encrypted to assure low intrusion, risks and privacy violation.
Low latency (relevant for OLTP and web applications)	Models can be tested off-line and tested before implementation. Updates are deployed into the model before it is released to the database
Scalability	Models are the first step to scalability. Models provide a database map to manage scalability for both single and multiple database platforms.

Table 1 Mapping DaaS Requirements to CA ERwin Data Modeler Functionality

This means that relational cloud prototypes should be designed starting with CA ERwin models. Both Customers and Providers have to synchronize shared model properties to satisfy the following relational properties:

- Implementing and sharing database structure models. This is the main way to better understand the challenges introduced into the database. Models should be implemented and deployed using a model-driven paradigm.
- Verifying database model properties according to private and public cloud requirements.
- Designing and testing new query types. Specific query classes need to support heterogeneous database environments. Typical examples are aggregation queries and text pattern matching queries. CA ERwin Data Modeler supports query design and testing to optimize database performance through multi-database structures.
- Designing of the data storage model. The model should enable query processing directly against databases to ensure privacy and secure changes from database providers.
- Modeling databases to calculate "a priori" physical resources allocation
- Modeling databases to predict usage "early" and to optimize database handling
- Designing multi-database structures to control and document pricing schemes (databases elasticity and scalability).

As a consequence, the diagram representing the high level relational cloud activities through CA ERwin Data Modeler is shown in Figure 1. CA ERwin Data Modeler defines service properties through which the DaaS process can be designed and maintained. The Relational Cloud is to be based on model prototypes that are shared, updated, secured and agreed upon among Customers and Providers: e.g. models are finally part of the cloud service pricing scheme

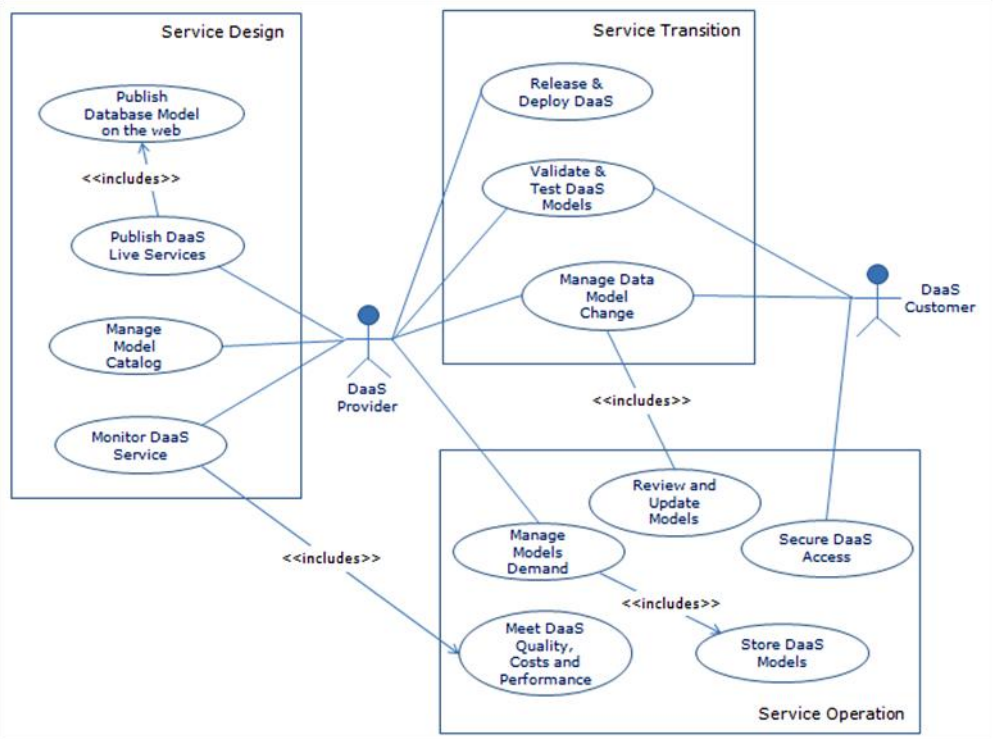

Figure 1

High level Use Case Diagram showing CA ERwin's Role in the Relational Cloud

CA ERwin in the Relational Cloud Architecture

Relational Cloud architecture should consider the following requirements:

Open Platform – This is a fundamental aspect of the CA ERwin Data Modeler architecture. Models can be deployed across database platforms without minimal restrictions. Database migrations are facilitated by using a common CA ERwin data model to translate the database-specific syntax from one platform to another.

Partitioning – CA ERwin data models can be deployed across different databases so partitions can be designed "ab initio" or created along the database life cycle. CA ERwin supports multiple partitions -- partitioned models can be further specialized by database partition as needed (agnostic partitioning). These actions are traced so that descendant partitions are linked to ancestors model by model.

Migration – Partitions can be migrated or moved through databases/platforms using model snapshots. Model snapshots protect changes that are made to database structures. This assists relocation of database partitions because maintenance and administration procedures can be performed off-line based on model properties. Models preserve aggregation queries and text pattern matching queries on data.

Resources allocation – One of the most critical aspects in DaaS is resources allocation. Models contain resources dimensioning and supply "as-is" and "to-be" scenarios. CA ERwin Data Modeler can estimate structure impact on database performance. Snapshot comparisons support model property optimization. This supports workload analysis metrics and resources allocation engines.

Scalability – Models can be scaled in independent partitions so scalability can be designed and tested before deploying databases structure across machines. The main value of scaling models is verifying "a priori" database consistency. CA ERwin Data Modeler can apply data warehouse models split by database/server. This preserves structure coherence and schemas.

Multi-tenancy – Deploying several identical databases in one or more machine and keeping track of usage and evolution consumes time and resources. CA ERwin Data Modeler can manage heterogeneous schemas belonging to a single model or different models. This assists in the multi-tenancy level of database deployment, aggregation consolidation and reallocation on the basis of model partitioning.

Unstructured storage – CA ERwin Data Modeler covers database storage on a model basis. Looking at database structure models, sub-models, partitions and snapshots maintain consistency of models deployed on different databases and/or machines. Database cloud storage can be governed by CA ERwin models that manage multiple changes through the cloud. Unstructured storage is also possible and CA ERwin Data Modeler governs evolution, aggregation and maintenance.

Figure 2 shows CA ERwin Data Modeler's role in the relational cloud architecture. Models satisfy partitioning, and replication and guide DaaS evolution through model change management and validation.

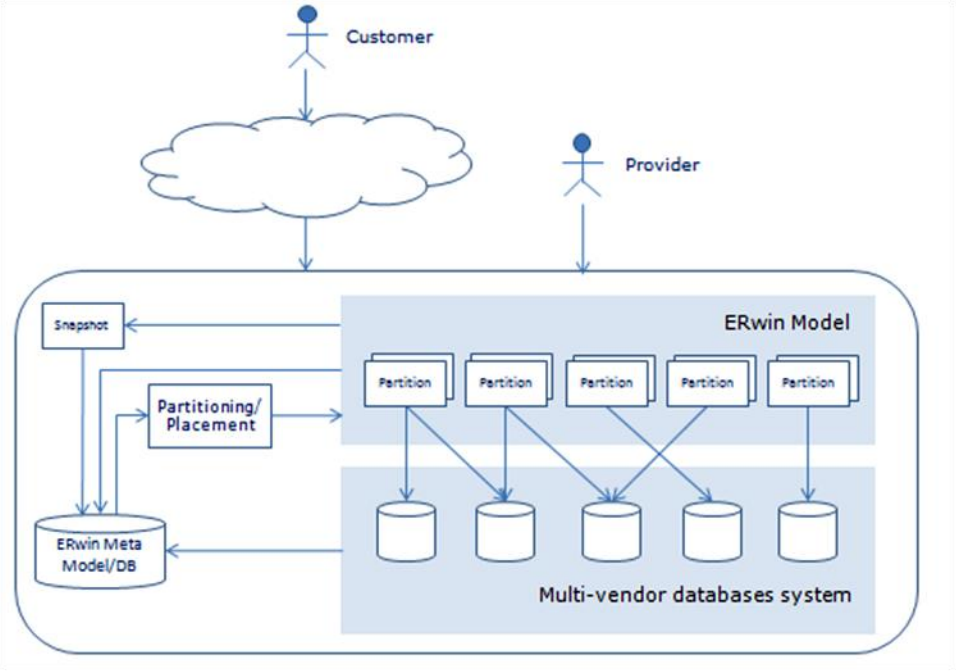

Figure 2

CA ERwin Data Modeler in the Relational Cloud Architecture

Providing Model as a Service: MaaS

Models can be shared, off-line tested and verified to define pricing schema, performance, placement and DaaS requirements. This means models themselves can be supplied as a service because customers have to verify how and why a database model meets their own data and application requirements. This aligns with the MaaS concept and it is completely supported by CA ERwin. As a consequence, CA ERwin MaaS should be the first Cloud Relational practice helping to tune the DaaS contract.

CA ERwin models, and generally speaking models as a practice (understanding before implementing and, overall, "know yourself"), govern design, deployment, storage, changes, resources allocation, and so on -- in other words what should be designed in terms of service. MaaS is a good practice for DaaS and should be extended to SaaS as well because in looking at application models, MaaS can guide implementation and define the framework with enough details to support projects plan understanding and development.

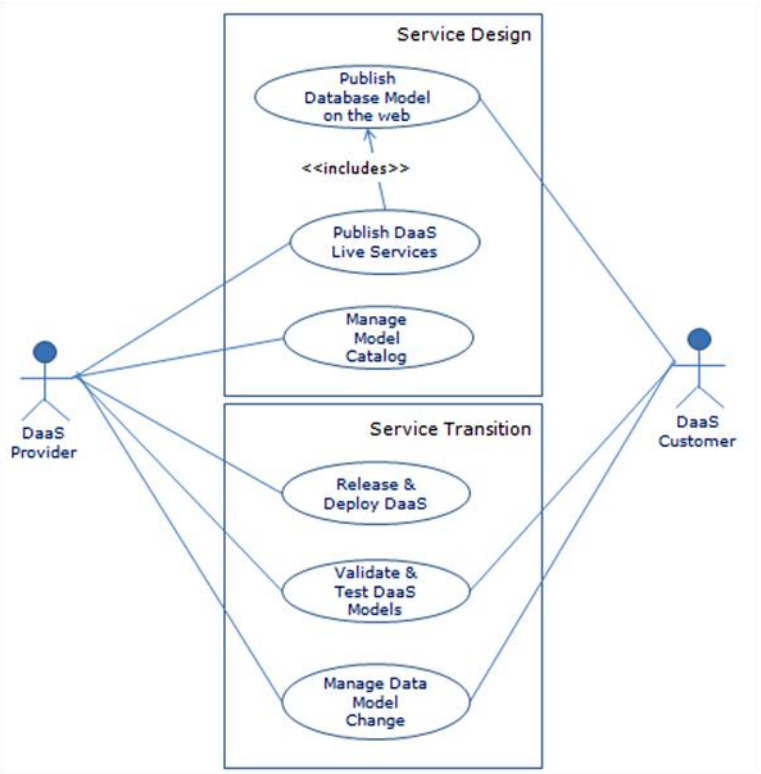

Figure 3
MaaS scenario

Models are published and can be modified and updated to meet customer system requests. Success models i.e. schemas that satisfy defined scenarios and optimize Relational Cloud algorithms create a DaaS template that can be shared and reused.

But who create models placed under MaaS? As shown in Figure 3, providers manage the knowledge updates to models to provide customer service. Customers generate an incremental improvement cycle to update aspects that can be shared among companies to create DaaS templates. CA ERwin users are privileged MaaS actors and can contribute to DaaS evolution through good practice models.

Conclusion

This paper introduces the role of CA ERwin in the Relational Cloud. Relational Cloud requisites have been analyzed and revised through CA ERwin Data Modeler. Relational Cloud architecture is taken into account and CA ERwin Data Modeler has been mapped to highlight how database models contribute to this architecture. Finally the concept of MaaS has been introduced, showing the model as the language assisting DaaS design and evolution.

CA ERwin and database models can add value to DaaS design and deployment. MaaS can open a new way to share DaaS information and tuning with CA ERwin Data Modeler acting as the knowledge server and the client to create, update and share DaaS properties.

Glossary

DaaS: Database as a Service;
MaaS: Model as a Service;
SaaS: Software as a Service;
SLA: Service Level Agreement.

Acknowledgements

I sincerely thank Donna Burbank for the precious feedback on contents and for encouragement on publishing this paper.

References

[1] N. Piscopo- ERwin® in the Cloud: How Data Modeling Supports Database as a Service (DaaS) Implementations http://erwin.com/uploads/whitepapers/ERwin_Cloud.pdf
[2] C. Curino et al. - Relational Cloud: a Database-as-a-Service for the Cloud
[3] M. Hogan – Cloud Computing & Databases, ScaleDB
[4] C. Curino et al. – Relational Cloud: the Case for a Database Service
[5] N. Piscopo – Modeling ITIL by UML, http://themodelgauge.spaces.live.com
[6] G. H. Sockut et al. Online reorganization of database – ACM Comput. Surv. 2009
[7] M. Hui et al. Supporting database applications as a service – ICDE 2009

About the Author

Nuccio Piscopo has more than 20 years in the IT industry as a Data and Application change management technical executive, with a University Degree in Physics, post lauream from the International School of Physics "Enrico Fermi", as well as a post lauream Master in Information Engineering (Computer Science). At present Nuccio leads the Technology and Consultancy Practice at Deveco and directs all customer-facing activities on delivering eGRC, Risk Management with special focus on virtualization and cloud audit risk assessment customer experience. In previous roles, he served at IR Semiconductors, Bachmann/Cayenne, Platinum Technology and CA Technologies. Nuccio Piscopo can be contacted through http://it.linkedin.com/in/nucciopiscopo.