ORACLE

Oracle Converged Database

Developing document-oriented applications with JSON, REST and SODA

Witold Świerzy
EMEA Core/Converged Database Expert

Agenda

- Introduction
- Oracle Converged Database – document oriented database
- MongoDB API Live Demo
- Summary

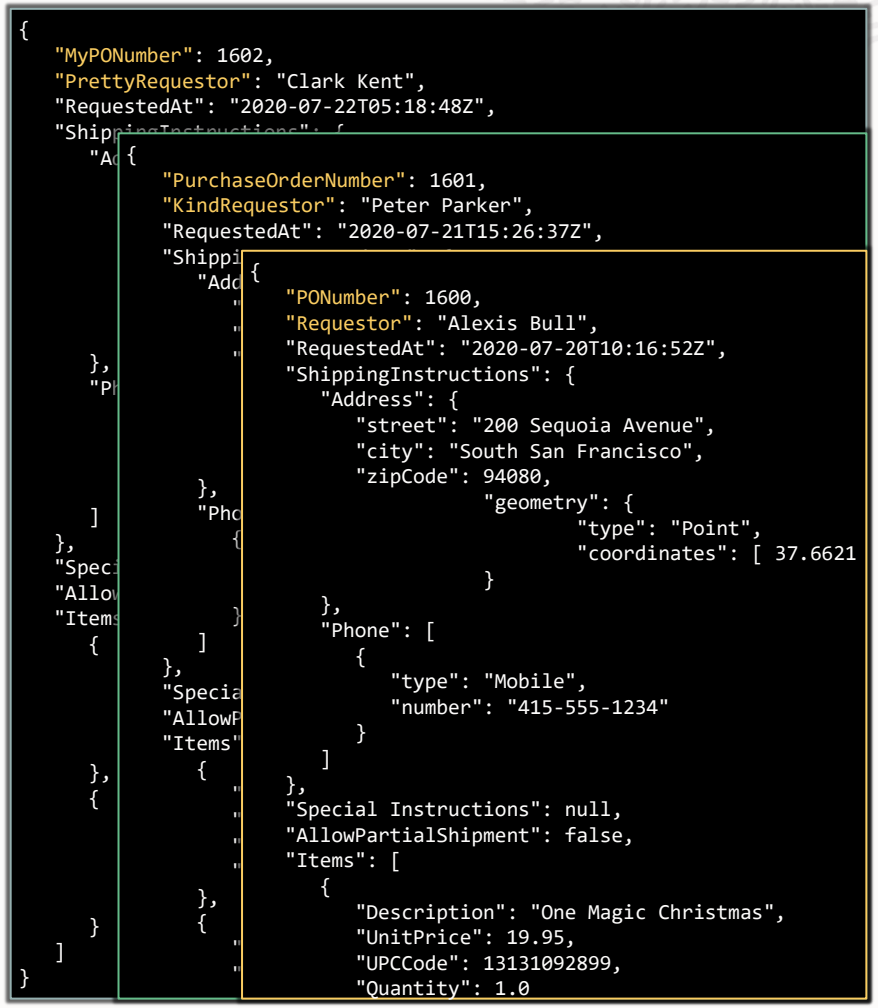
Introduction

Modern Applications need flexible schema

Mobile applications	**Internet of things**	**360 degree customer view**	**User profile management**	**Catalog data**
				
Content management	**Online advertising**	**Real time Big data**	**Social network**	**Gaming**
				

Why JSON can be an answer ?

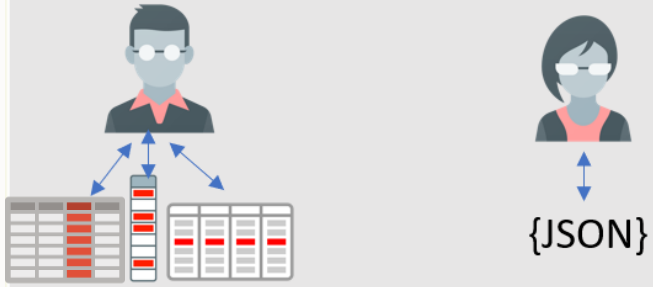
- JSON stands for JavaScript Object Notation
  - Defined by standards ECMA-404 and ECMA-262
- JSON documents have
  - Fields
  - Values (Strings, Numbers, Booleans, ISO 8601 Dates, Nulls, Objects, Arrays)
- JSON documents are stored in Collections
  - Each document may have **its own Schema**
  - Schemas are flexible



Why JSON can be an answer ?

1: JSON easily maps to Business Objects

- no need to map object to multiple tables
- single insert/query/update
- less complex SQL



2: JSON is schema flexible

- adding additional fields
  - *add a middle name or birth name*
- cardinality change
  - *1 address -> multiple addresses*
- Productivity, Agility

```
{
 "firstName" :"John",
 "middleName":"Jeremy",
 "lastName"  :"Smith",
 "age":25,
 "address":[ {…}, {…}]
```

3: JSON Document Store APIs

{APIs}

- Avoid SQL for simple (CRUD) operations
  - insert/find/update/delete a person
  - noSQL-style document APIs

```
SODA get –f {"lastName":"Smith"}
```

versus

```
SELECT id, data
FROM persons
WHERE data.lastName = 'Smith'
```

- Preserve SQL for reporting, analytics, ML

Customer: Major US Insurance:

- every insurance policy is a JSON doc
- different policy shapes
- all policy information in one doc
- no decomposition/joins
- low latency puts/gets
- great for web-application

- Relational views
  - reporting
  - daily risk analysis

Customer: Major Global Stock Exchange

- metadata for financial products
- 1600 possible attributes
- 100 used on average
- application adds new attributes if needed without database op
- attributes in JSON column

- Custom reporting for banks
- SQL/JSON operations

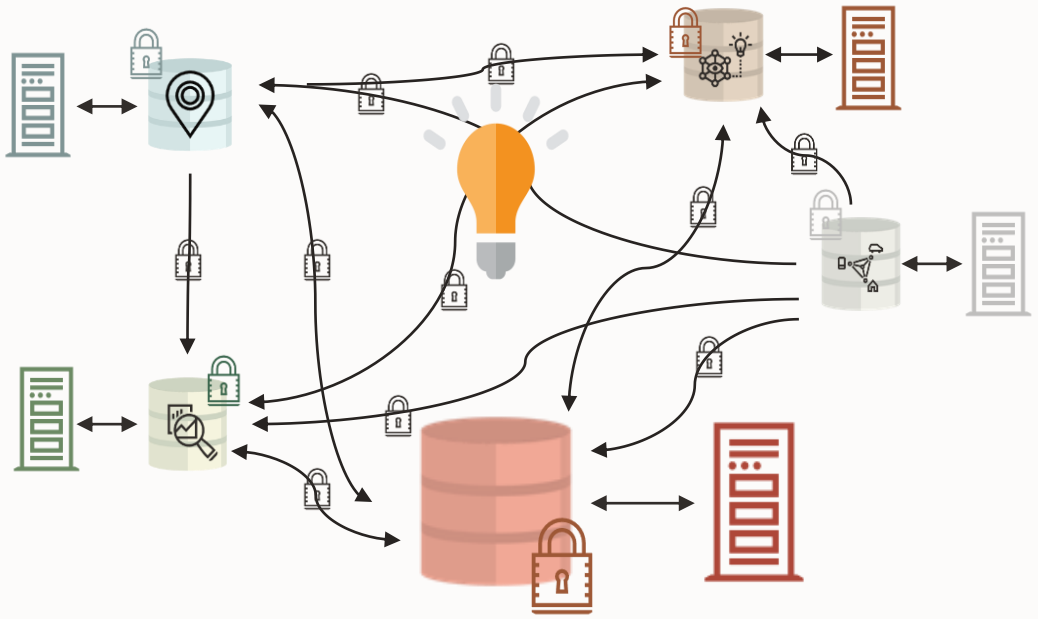
Customer: Global Retailer, stores and online shop

- Point of sale application
  - order status tracking
  - payment tracking
  - gift card processing
- Developers preferred document store model
  - agile, productive, easy to use
- DBA and Ops preferred Oracle
  - transactions
  - SQL for reports
  - security features,…

NoSQL

SQL

Multiple single-purpose databases may drive to complex and unmanageable architecture



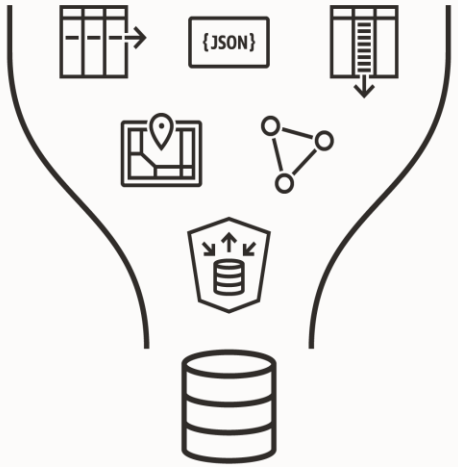
ETL, replication, events etc.

Traditional paradigm assumes using many single-purpose databases drives to

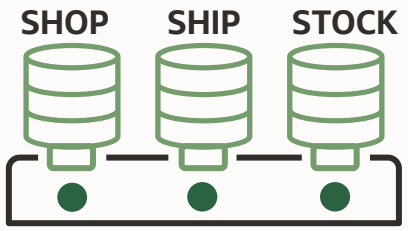
- High level of complexity of maintenance and development process
- High TCO Costs

Oracle Database

A Converged, Open SQL Database



Converged Database



Support for microservices

Multi-model

Best-of-Breed Relational, JSON, Spatial, Graph, Cube, Text, Blockchain
Cross-model operations enables you to easily create value across all your data

Multi-workload

High Performance Transactions, DW, Analytics, ML, IoT, Streaming, Multitenant
Deep optimizations deliver exceptional price-performance across all workloads

Most productive for developers and analysts

Same SQL and transactions operate on any data and workload
Integrated microservices, events, REST, CI/CD, Low-code

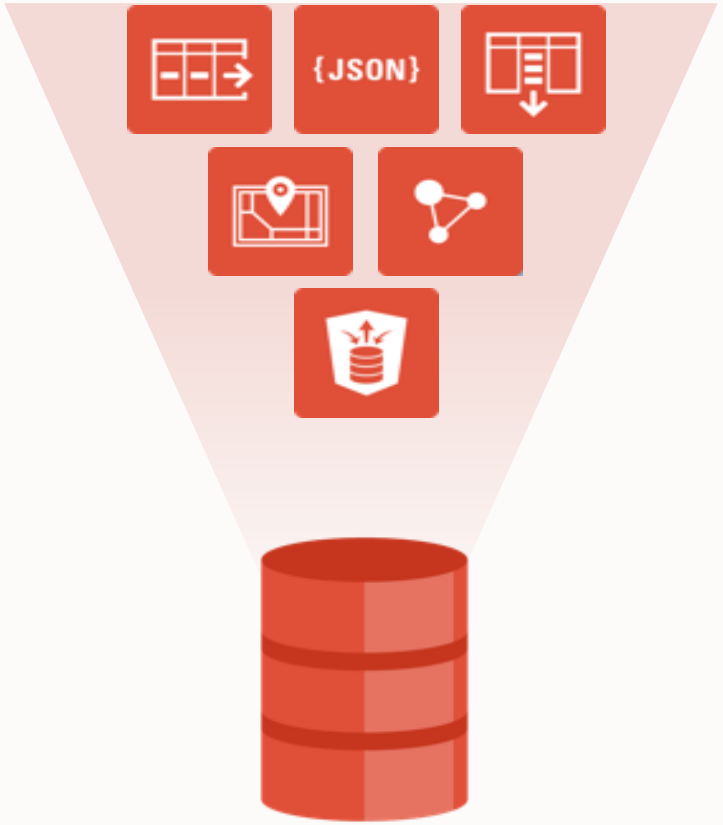
Support for microservices

Low-level data types and workloads should not dictate your architecture

Oracle Converged Database
Document-oriented database

Oracle Converged Database

support for all modern Languages/Drivers/Tools/APIs



Languages

- C
- C++
- Java
- .Net
- JavaScript
- Python
- PHP
- R
- Go
- Rust
- Ruby
- Perl

Tools
Code Versioning

- SQL and PL/SQL
- SQL Developer, Data Modeler
- VS Code plugin
- SQLcl (modern sqlplus)
  - With *Advanced* Liquibase

Modern APIs

Collections API

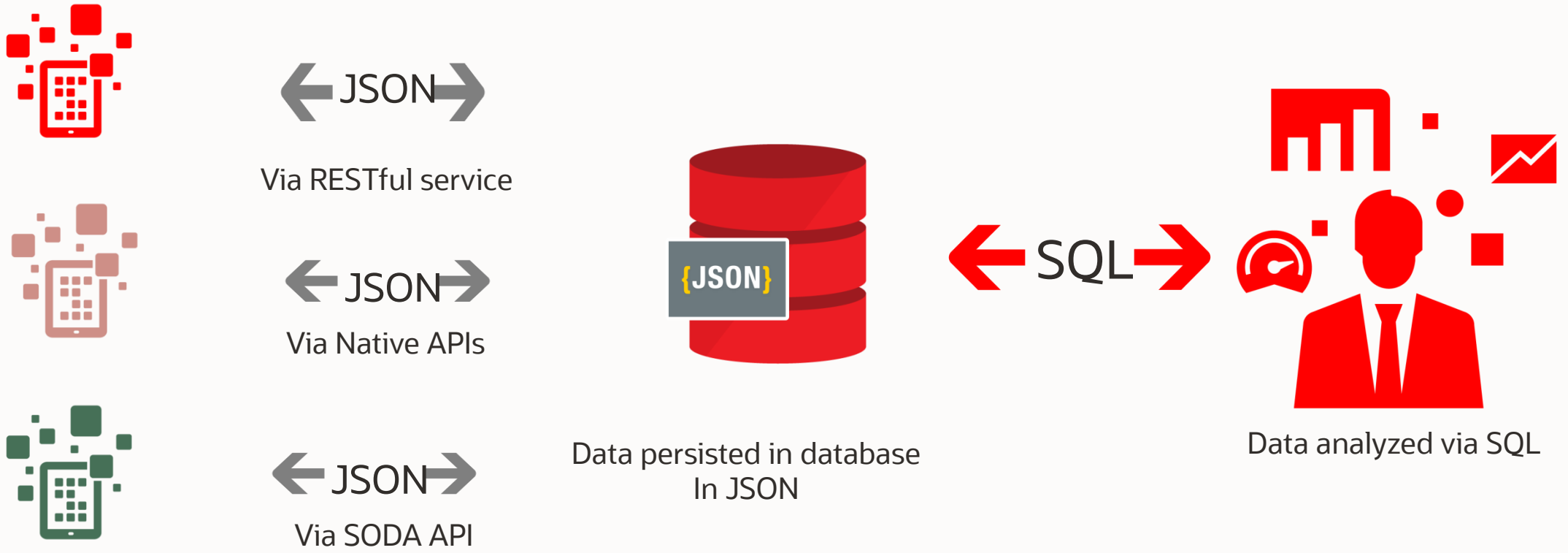
- SODA drivers
- JSON results

Data As a Microservice with REST API

- JSON Data through HTTPS

Oracle Converged Database

Access JSON data via multiple API(s)



Oracle Converged Database

- Full SQL support
- ACID transactions
- JSON search indexes
- Advanced security
- APEX low-code development

Oracle Converged Database

- You have the choice to store JSON data as
  - Collections
  - Column(s) in relational tables

- JSON documents are stored as
  - Native Binary data type
    - **JSON** (up to 32 MB)
  - Textual data type
    - VARCHAR2 (up to 32* KB)
    - BLOB (up to 2 GB)
    - CLOB (up to 1 GB)

Support for MongoDB

Develop and run MongoDB workloads in the Oracle Autonomous JSON Database

Modern document-centric development

- JSON Collections-based data model
- Rich clients – MongoDB API, REST and SODA based development API
- Native JSON storage with advanced indexes and optimized performance

... and proven enterprise functionality

- ACID Transactions
- SQL-based Reporting and Analytics (including scalable parallel execution)

... running on Oracle Database platform

- Availability
- Security
- Elasticity

Support for MongoDB

Develop and run MongoDB workloads in the Oracle Autonomous JSON Database

API compatibility details

MongoDB API version 4.2 compatible

- Transactions supported
- Load balanced connections supported like on MongoDB Atlas

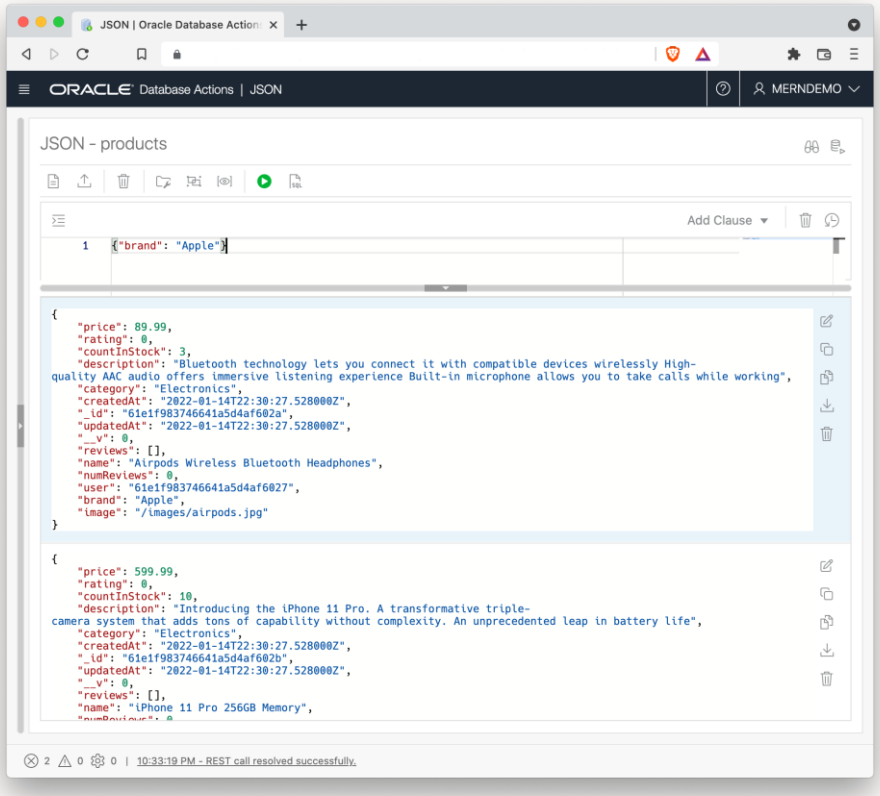
MongoDB API backed by Oracle user and roles management instead of MongoDB's

- Allows Oracle's enterprise-class security features to be used with MongoDB collections
- Makes unified user management easy

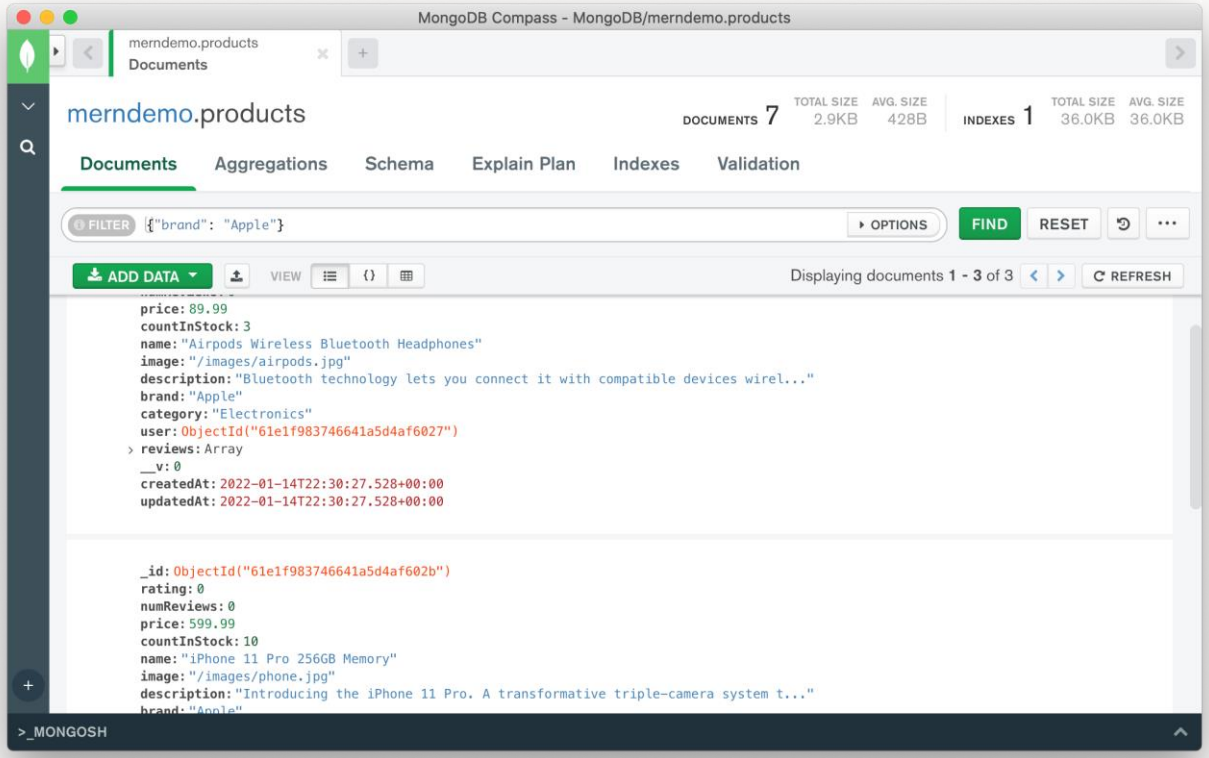
Oracle Autonomous Database - support for MongoDB

Develop and run MongoDB workloads in the Oracle Autonomous JSON Database

JSON Development IDE



Oracle Cloud tools



MongoDB Compass and other tools

Support for MongoDB

Recommended tools and driver versions or higher with support for load-balanced connections

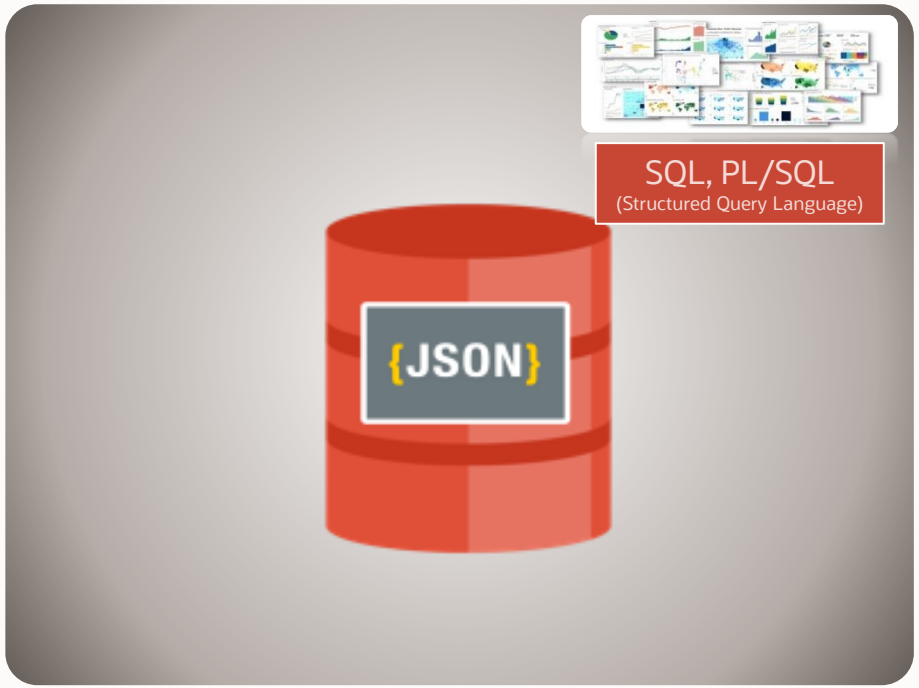
- C 1.19.0
- C# 2.13.0
- Compass 1.28.1
- Database Tools 100.5.0 (includes mongoexport, mongorestore, and mongodump)
- Go 1.6.0
- Java 4.3.0
- MongoSH 0.15.6
- Node.js driver 4.1.0
- PyMongo 3.12.0 (for Python language)
- Ruby 2.16.0
- Rust 2.1.0

Source: https://docs.oracle.com/en/database/oracle/mongodb-api/mgapi/overview-oracle-database-api-mongodb.html#GUID-9B704340-5E1D-4CA9-8944-6296A69221B0

Oracle Converged Database

SQL support for JSON

- SQL for advanced reporting and analytical operations on JSON documents
  - Dot notation
    - SELECT coll.doc.address.postalCode…
  - SQL/JSON Path Expressions
  - Rich set of functions
    - Views + On-Demand Refreshable Materialized Views
    - Analytical functions (over (), window ()…)
    - Documents Pattern Matching (match_recognize)
    - JSON documents generation from relational data
    - …
- PL/SQL for server-side processing
  - Parse and generate Documents
    - apex_json
  - Secure access to JSON Open Data
    - apex_web_service

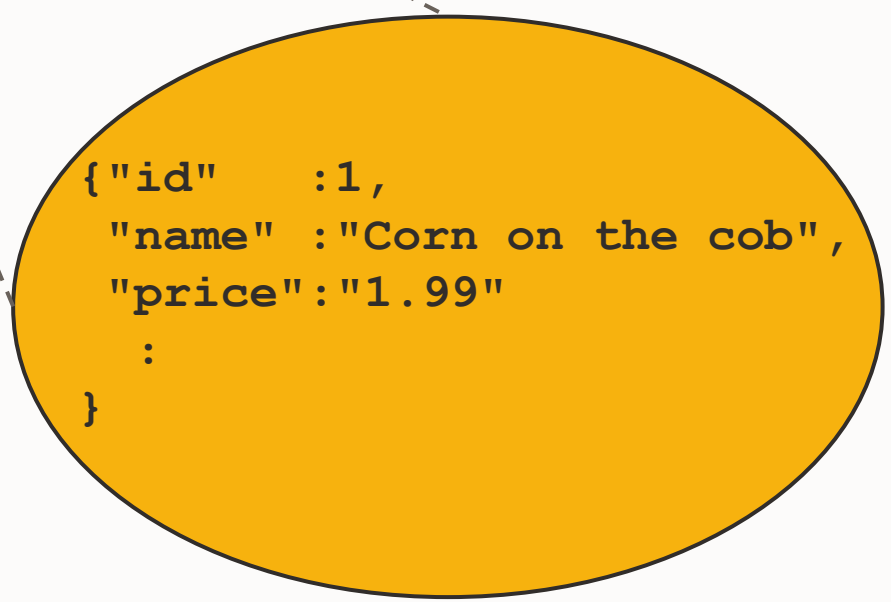


Oracle Converged Database

SQL support for JSON

```
-- 1. Create a menu-items table
-- where menus items are stored as JSON

CREATE TABLE menu_items(
   rest_name   VARCHAR2(255),
   menu-item    JSON);
```




```
-- 2. Use simple dot notation to access
-- elements within the JSON docs

SELECT m.menu_item.name  item,
       m.menu_item.price price
FROM   menu_items m;

ITEM                 PRICE
-----------          ------
Corn on the cob       1.99

-- 3. Create index on a JSON column
-- element to speed up rows filtering

  CREATE INDEX menu_items_idx ON
  purchase_orders
  ( JSON_VALUE(JSON_DOCUMENT, ,$.id.number()'
  ));
```

Oracle Converged Database

SQL support for JSON

Native Support to Index and Search JSON Documents

Oracle makes it simple for Apps to index, search and analyze text

Search text using keyword search, context queries, pattern matching, etc.

- Search websites, catalogs, documents, LOBs

Perform linguistic analysis on documents to easily classify them

- Classify customer feedback as positive, negative, or neutral using sentiment analysis

Search JSON or XML documents using the structure of the document to restrict the search

- Find all purchase orders where the comment field contains “damage” and “delivered”

Oracle Converged Database

SQL support for JSON

Native Support to Index and Search JSON Documents

```
-- 1. Create a text index on the JSON column of the purchase_orders table

CREATE SEARCH INDEX PO_search_ind ON purchase_orders(po_doc) FOR JSON;


--2. Search for PO documents with “Delivered” and “Damage” in the comments
field

SELECT po_number, po_doc.comment comment
FROM   purchase_orders
WHERE  JSON_TEXTCONTAINS(po_doc, '$.comment', 'Delivered and Damage');
```

Oracle Converged Database

SQL support for JSON

Ensuring consistency

- Constraints

```
CREATE UNIQUE INDEX unique_idx ON purchase_orders
( JSON_VALUE(JSON_DOCUMENT, '$.PONumber.number()' ));

SQL> soda insert purchase_orders {"PONumber": 1601 };
SQL> commit;

SQL> soda insert purchase_orders {"PONumber": 1601 };

ORA-00001: unique constraint (HR.UNIQUE_IDX) violated
```

```
ALTER TABLE purchase_orders ADD CONSTRAINT
po_address_zipcode_positive_number CHECK (
   JSON_VALUE( json_document,
               '$.ShippingInstructions.Address.zipCode'
               returning number)
> 0 )
ENABLE      -- enable constraint
NOVALIDATE; -- but don't check existing documents
```

- Triggers

```
CREATE TRIGGER validate_items
AFTER UPDATE ON purchase_orders FOR EACH ROW
DECLARE l_number_of_products number;
BEGIN
    SELECT sum(Quantity) into l_number_of_products
      FROM JSON_TABLE(:NEW.json_document, '$'
          Columns( Nested Items[*] Columns(Quantity NUMBER)));

    IF l_number_of_products > 10 THEN
        RAISE_APPLICATION_ERROR(-20001, 'No more than 10
            products ordered at the same time.');
    END IF;
END;
/

SQL> UPDATE purchase_orders SET json_document =
        JSON_TRANSFORM( json_document, APPEND '$.Items' =
'{"Description":"Shiny toy", "UnitPrice":1.0,
  "UPCCode": 123456789, "Quantity": 5}' format json );

ORA-20001: No more than 10 products ordered at the same time.
```

Oracle Converged Database

SODA (Simple Oracle Document Access) and SQLcl

SODA: Goals

- Enable schemaless development on top of an Oracle Database
  - Provide a simple NoSQL-style API for working with documents
- Make it easy to use Oracle as a NoSQL-style document store
  - Allow developers to work with Oracle without learning SQL
  - Allow a developers to work with Oracle without DBA support
- Support all common application development environments
  - Traditional programming languages
  - Scripting languages and frameworks

Oracle Converged Database

SODA (Simple Oracle Document Access) and SQLcl

SODA APIs

- NoSQL-style APIs for
  - Java, JavaScript/Node.js, Python, REST, PL/SQL, C…
- Used to manage JSON data
  - create collections
  - store documents in collections
  - retrieve documents
  - query documents
  - ***No need to know SQL!***

SQLcl

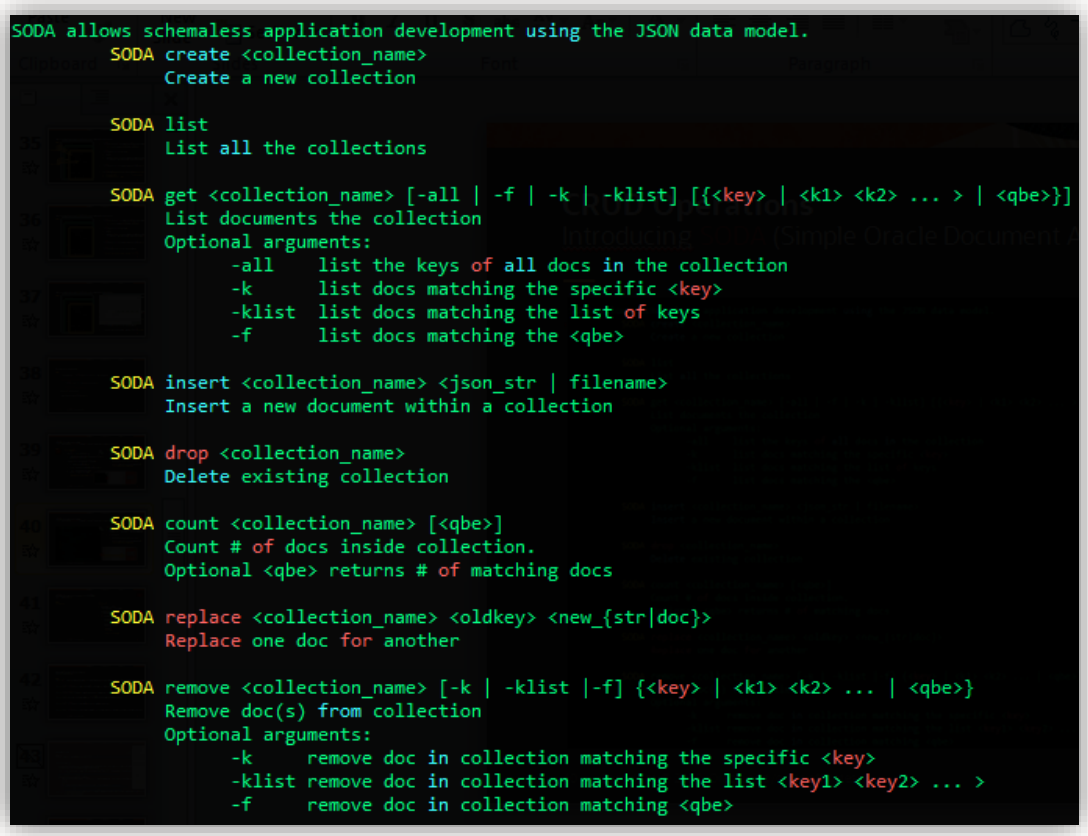
- Modern SQL Developer Command Line interface for Oracle database
- Provides
  - inline editing, statement completion, command recall…
  - **SODA commands**

Oracle Converged Database

SODA (Simple Oracle Document Access) and SQLcl

SQLcl

- Modern SQL Developer Command Line interface for Oracle Database
- Provides
  - Inline editing, statement completion, command recall…
  - **SODA commands**
    - **Full CRUD operations support**



Oracle SODA

examples

Node.js

```
conn = await oracledb.getConnection(…);
db = conn.getSodaDatabase();
col = await db.createCollection("purchase_orders");
await col.drop();
```

Java

```
OracleClient client = new OracleRDBMSClient();
db = client.getDatabase(jdbcConn);
OracleCollection col =
db.admin.createCollection("purchase_orders");
col.admin().drop();
```

Python

```
conn =cx_Oracle.connect(…);
db = conn.getSodaDatabase();
col = db.createCollection("purchase_orders");
col.drop();
```

PL/SQL (and Oracle Application Express)

```
col := dbms_soda.create_collection('purchase_orders');


select dbms_soda.drop_collection('purchase_orders')
from dual;
```

Oracle SODA

Sample services provide by SODA for REST

GET /DBSODA/schema	List all collections in a schema
GET /DBSODA/schema/collection	Get all objects in collection
GET /DBSODA/schema/collection/id	Get specific object in collection
PUT /DBSODA/schema/collection	Create a collection if necessary
PUT /DBSODA/schema/collection/id	Update object with id
POST /DBSODA/schema/collection	Insert object into collection
POST /DBSODA/schema/coll?action=query	Find objects matching filter in body

Oracle REST Data Services

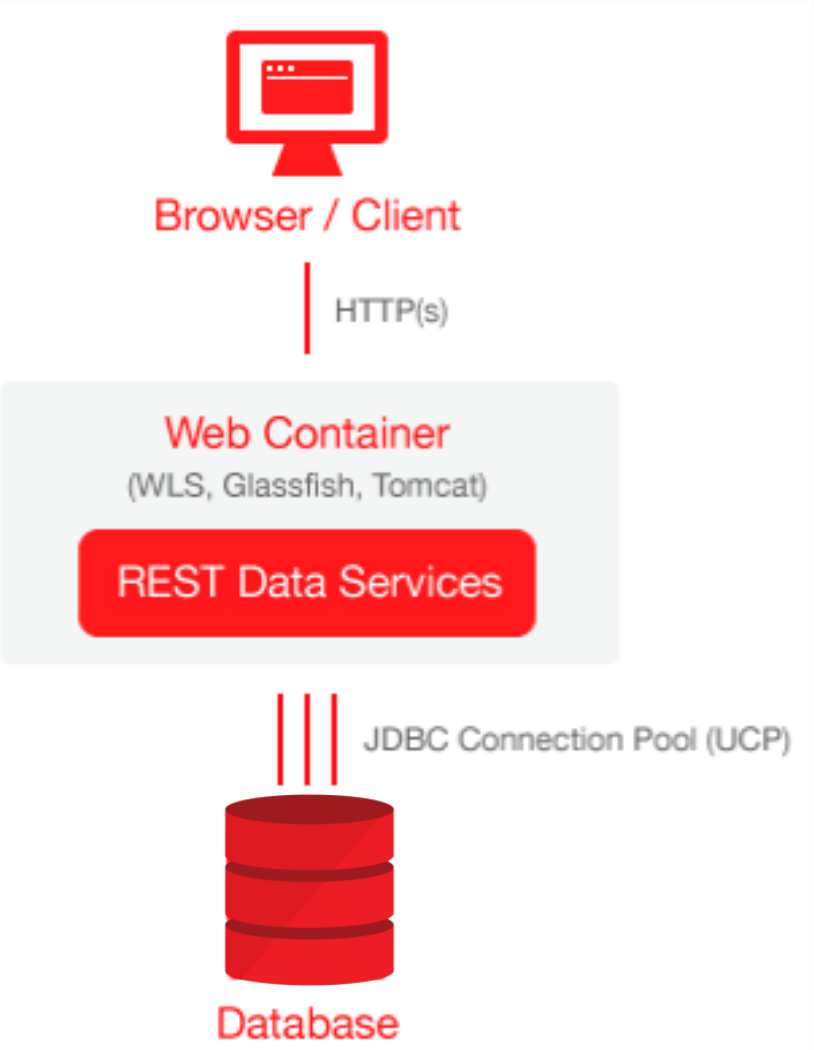
ORDS powers the Schema Service, A Quick Detour

- Provides data access consistent with modern App Dev frameworks
  - Mid tier application
  - Can map standard http(s) RESTful gets and posts to SQL
  - Can declaratively returns results in JSON format
  - JavaScript friendly
  - Can support high numbers of end users
- Services
  - HTTP(s) relational data access
  - Oracle JSON collection based schema-less access
  - Oracle NoSQL access over HTTP
  - Oracle APEX mid-tier, web toolkit applications, mod_plsql replacement

Oracle Converged Database

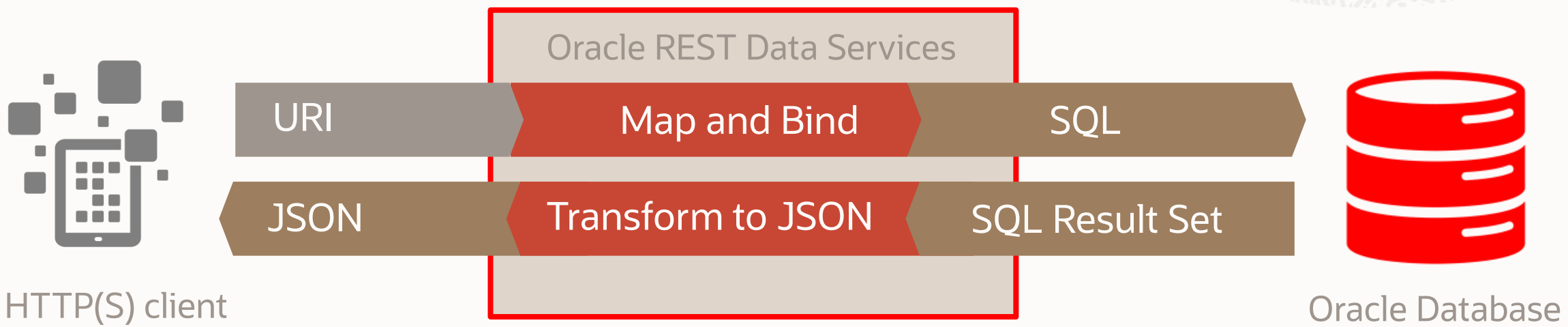
Oracle REST Data Services

- Available Standalone (Jetty), Weblogic, Tomcat & Glassfish
- Turns Database Service into an RESTFul API service
- Fully provisioned and functional in all cloud editions
- Available since 11g, no extra cost
- Allows publishing of URI based access to Oracle database over REST
- Results in JSON or CSV
- Mapping of URI to SQL or PL/SQL
- All HTML methods GET, PUT, POST, DELETE, PATCH
- Oauth2 integration
- Highly scalable, can use all feature of database



Oracle REST Data Services

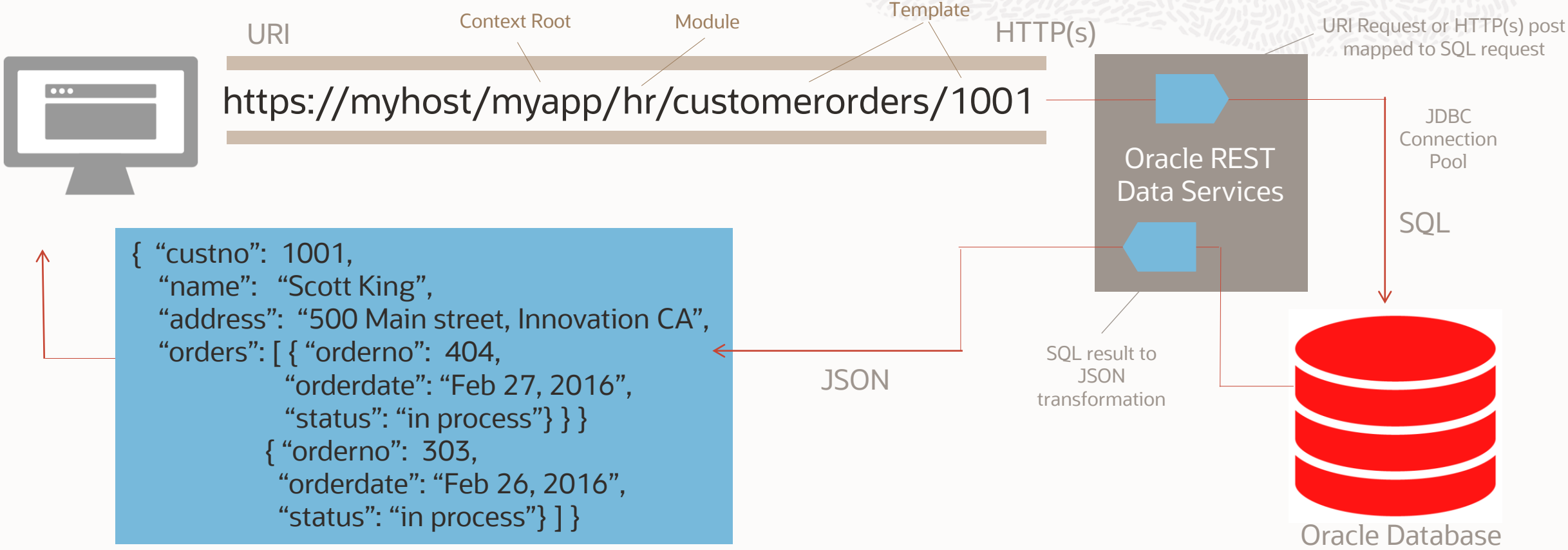
Serving JSON results from relational data



- Data stored in standard relational tables and columns
- Oracle REST Data Services (ORDS) Developer defines URI<>SQL mapping
- App Developer calls named URI over HTTP(S) gets and posts

Oracle REST Data Services

HTTP(s) API App-Dev with Relational Tables in Oracle Database



ORDS maps standard URI requests to corresponding relational SQL (not schemaless): e.g. SQL SELECT from customers and orders table.
ORDS also transforms the SQL results into JavaScript Object Notation (JSON), other formats include HTML, binary and CSV.
Fully committed to supporting any and all standards required by Fusion / SaaS / FMW; we are actively engaged in the ongoing dialog.

Oracle REST Data Services

RESTFul development with JSON

RESTFul services are a simple well, understood model

CRUD operations map to HTTP Verbs

Create/Update : PUT / POST
Retrieve:GET – Delete : DELETE

Other operations, such as Query by Example, Bulk Insert and Indexing are mapped to variants of POST

JSON document forms the payload of the HTTP Request or Response

Stateless model, no transaction support

Oracle Database

The most advanced database platform in the market

Document-oriented data model is fully supported by all the database options

Consolidated and Mixed-Workloads

Multitenant Database, Exadata, OLTP and analytics in one

Fault-Tolerant Availability

RAC, Data Guard, Flashback, Online Data Reorg, Online Patching, MAA

Highest Security

Data Safe, Advanced Security, Encryption, Roles, Privileges …

Transparent Scaling

RAC, Exadata, Parallel SQL, Native Sharding

Comprehensive Data Model Support

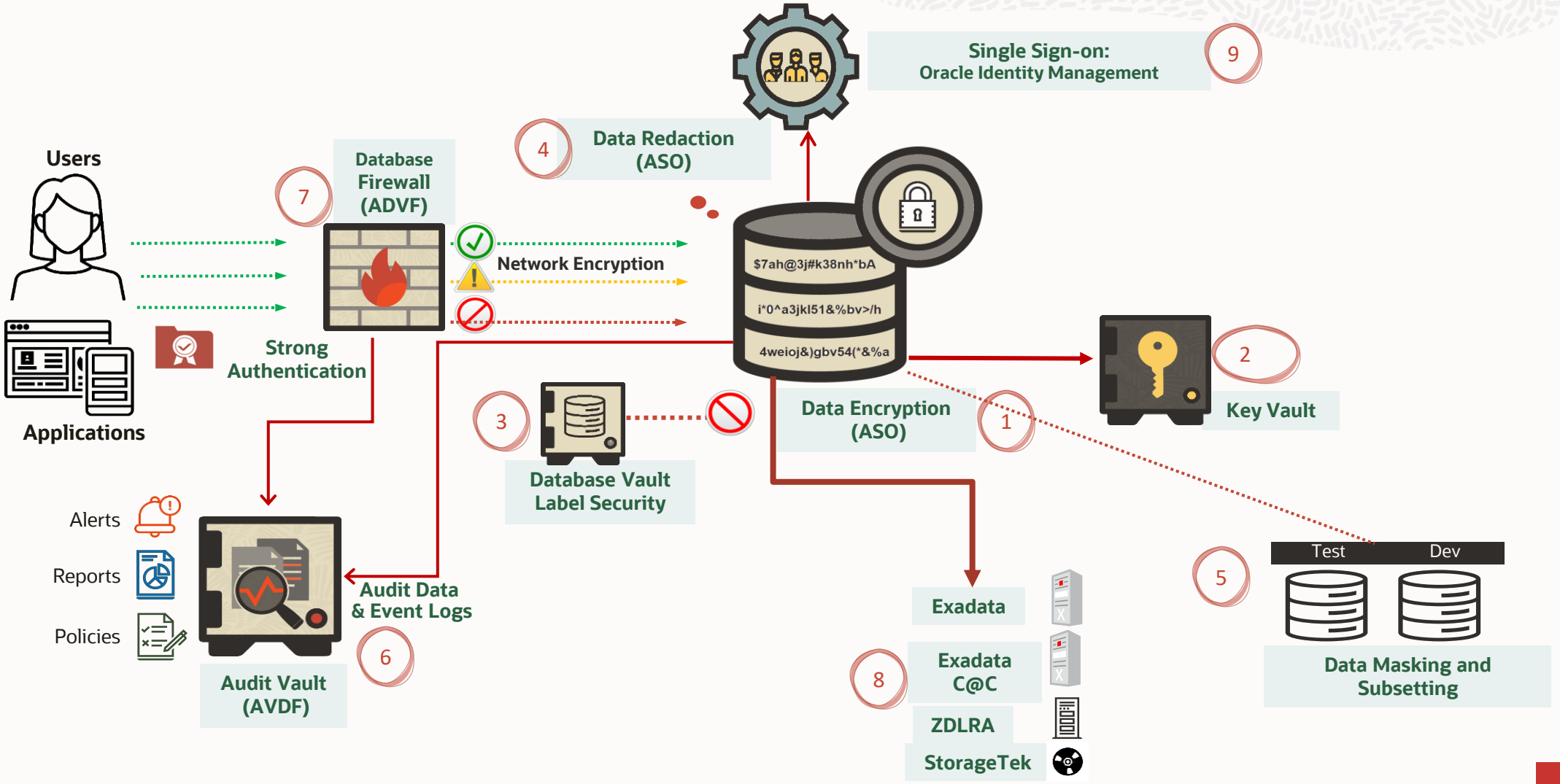
JSON, text, graph, spatial, blockchain

Powerful Analytics

In-memory analytics, machine learning, data partitioning

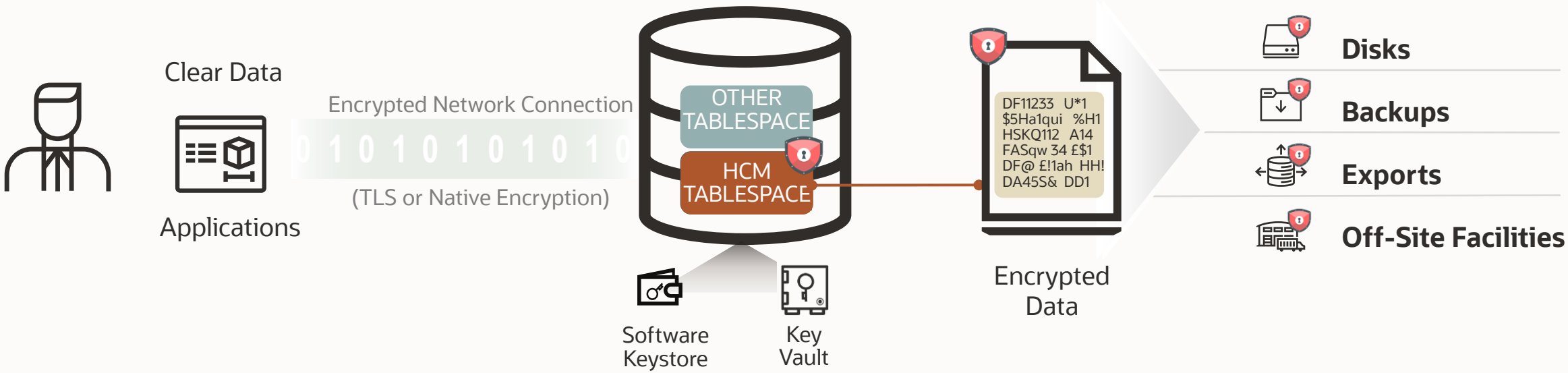
These technologies fully support document-oriented Oracle Converged Database

Oracle's unique approach: Maximum Security Architecture



Oracle Converged Database

Transparent Data Encryption (TDE)



Encrypts entire application tablespaces or an application column

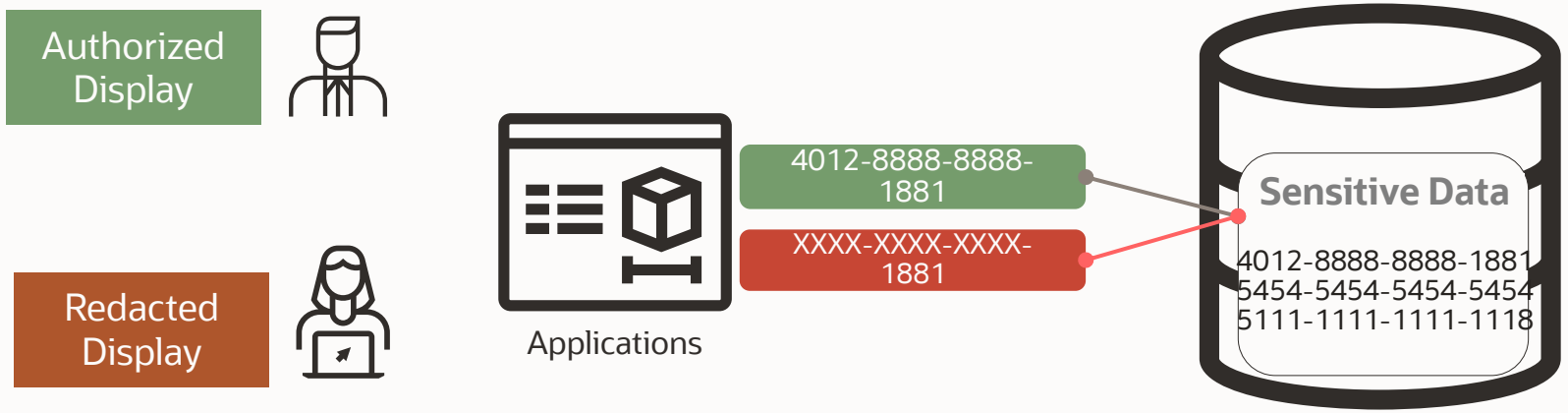
Protects the database files on disk and in backups

No application changes required

Integrated with the Oracle technology stack

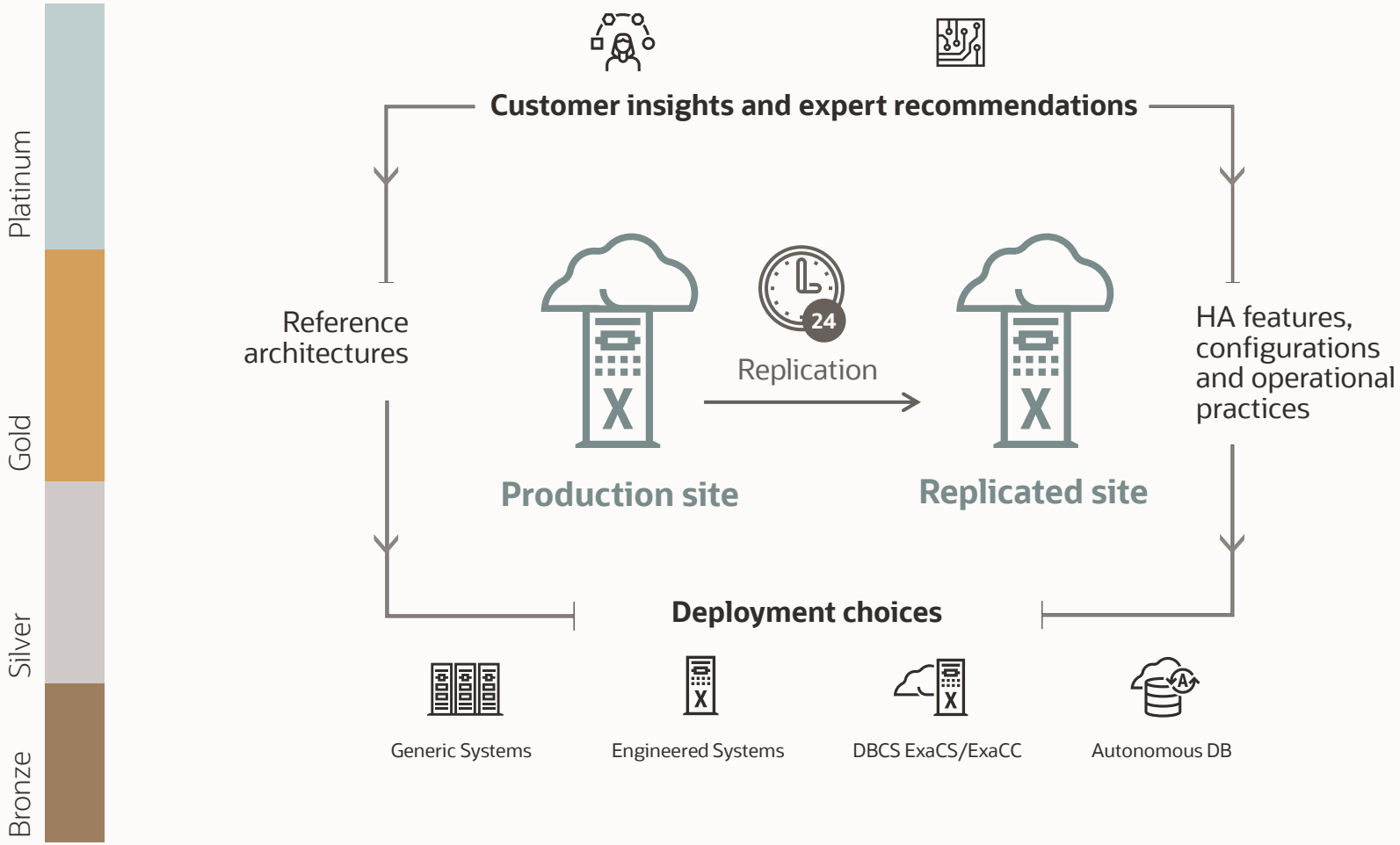
Oracle Converged Database

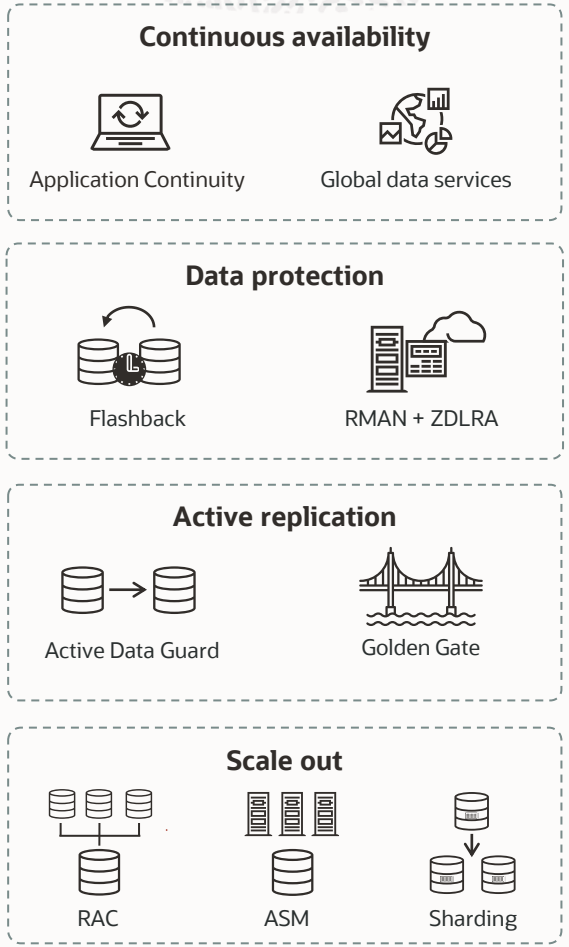
Data Redaction (Part of Advanced Security Option)



- Dynamic masking of application data based upon user name, IP, application context, and other session factors
- Library of redaction policies and point-and-click policy definition
- No impact on operational activities

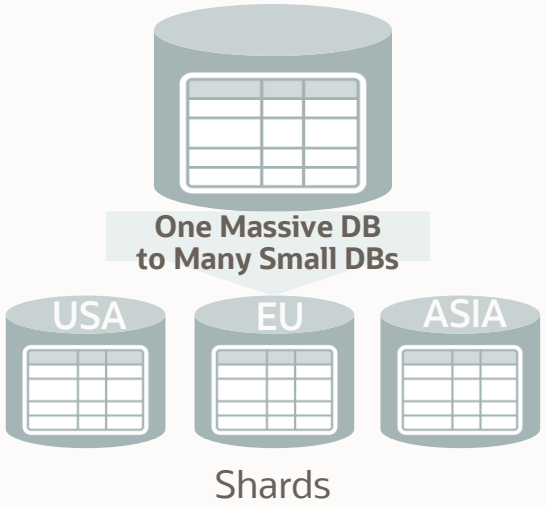
Oracle Maximum Availability Architecture (MAA)





Oracle Converged Database

Native Sharding Simplifies Distributed Data Architecture



Oracle makes it simple for Apps to deliver Data Sovereignty or Massive-Scale using native Sharding

Shard monolith databases into a farm of smaller databases

- Shards can be placed in-country to satisfy data sovereignty
- Shards are fully isolated **- linear scalability of data and users**
- Routes SQL based on shard key, or runs cross shard SQL
- Online addition and reorganization of shards

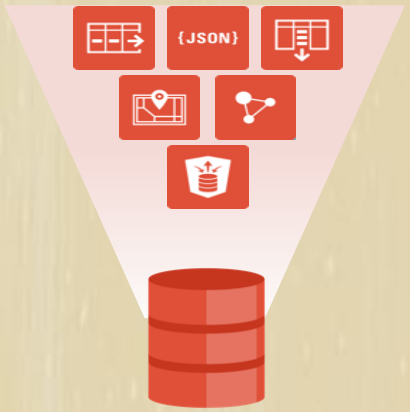
All the benefits of sharding with all the benefits of a mature SQL Database

MongoDB API Live Demo

Live Demo of MongoDB API 23c is fully described in

Oracle Converged Database 23c MongoDB API Live Demo.pdf file

Summary



Modern applications need to generate value from data in new ways

- They are built using new development methodologies and technologies
- But it complicates Database Architecture by using multiple single-purpose databases
- Development focused on Integration

Oracle Database allows for easy start developing document-oriented applications

- Provides multiple APIs
- Fully supports document-oriented data model based on JSON specification
- Provides MongoDB API for MongoDB applications
- Provides multiple enterprise-scale solutions, which increase its availability, security and performance

Some Developer Resources for free

- Oracle Database Express Edition (Version 21c)
- Pre-Built Oracle DB Developer VM
- Docker Support
- Live SQL: without installation learn and share http://livesql.oracle.com)
- Autonomous Database Free Trial
- Free SQL courses:
  Databases for Developers: Foundations