Journey to Oracle Utilities Customer to Meter

Migrating, Upgrading and Merging
Customer Care Billing (CCB) and Meter Data Management (MDM) to Customer to Meter (C2M)

Technical Overview

ABJAYON WHITE PAPER | NOV 2025

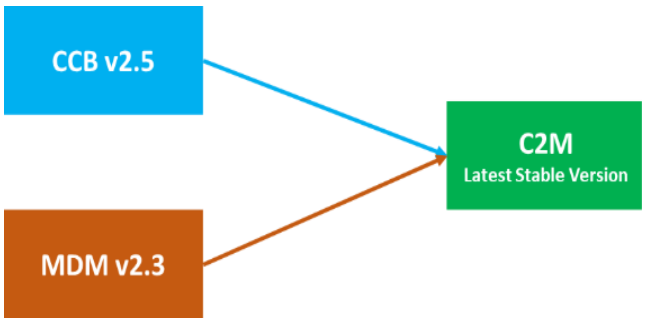
Executive Summary

Oracle Utilities Customer Care and Billing (CC&B) is an enterprise application designed to manage customer information, billing, and revenue for utility companies — including water, electricity, and gas providers whereas Oracle Utilities Meter Data Management (MDM) is a core component of the Oracle Utilities product suite. It is designed to collect, validate, store, and process high volumes of interval and scalar meter data from smart meters and other devices. Oracle Utilities Customer to Meter (C2M) unifies CC&B and MDM into a single platform, reducing integration overhead and improving data consistency.

Oracle Utilities Customer to Meter (C2M) is necessary for utility companies seeking a unified, efficient solution that bundles **customer information, meter data management, and billing** into a single system. This integrated approach streamlines the entire "meter-to-cash" process, reducing operational complexities and enhancing customer service.

This whitepaper outlines the business need to integrate Customer Care Billing and metering solution into single integrated Oracle Utilities Customer to Meter, and there by benefitting from the single integrated and feature rich application. The white paper also highlights high level solution approach, key considerations and challenges typically we have seen upgrading oracle customer care billing and meter data management into Oracle utilities customer to meter solution.

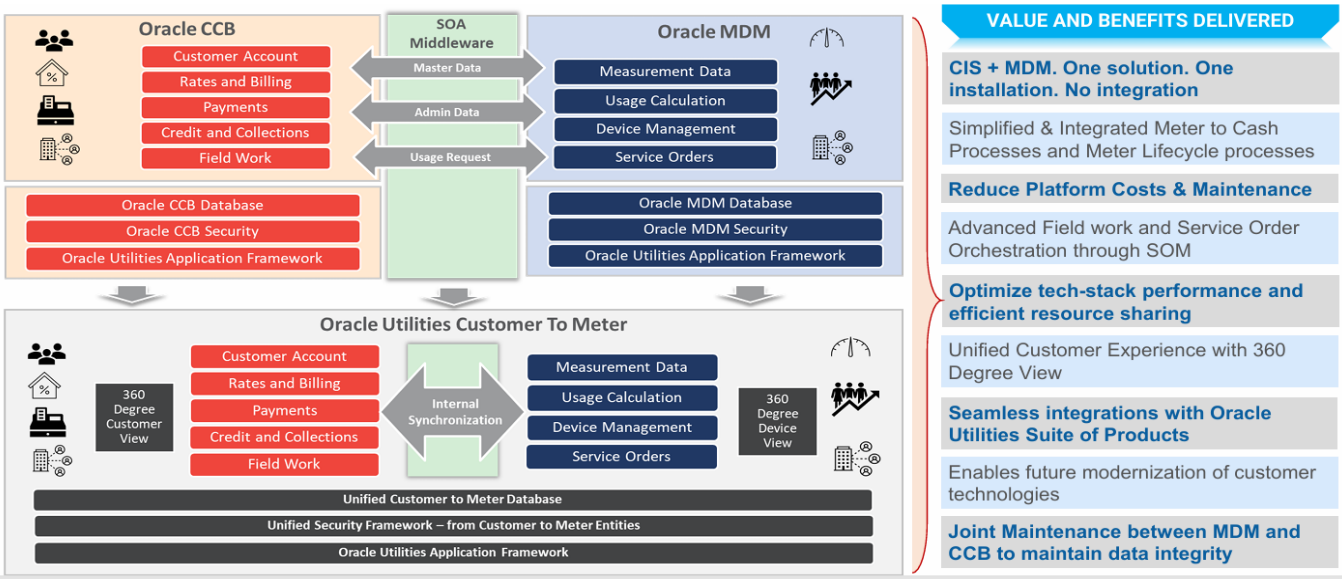
Although, the whitepaper is targeted towards the technical upgrade approach and implementation best practices, it is important to highlight the approach should be tailored towards the utility and customer objectives – including the feasibility and compatibility factors from learnings of the existing systems landscape.



Benefits and Value Add

The necessity of bringing Oracle Customer Care Billing - Customer Information and Oracle Utilities Meter Data Management - Meter Operations products into single umbrella of Customer to Meter stems from several critical needs within the modern utilities industry and utilities driving business factors.

- **Enhanced Customer Experience:** C2M Provides a 360 Degree View of both customer and meter information in a single dashboard view, enabling much faster, highly insightful customer service interactions. It supports online account management, self service touchpoints and multi-channel communication, meeting digital customer demands for data access and flexible service controls.
- **Elimination of Complex Integrations:** C2M Runs on a single technology platform and one database, removing the need of complex middleware integrations between Customer Care Billing and Meter Data Management Systems, and managing the continuous synchronization of the master data between these two systems. This reduces Total Cost of Ownership (TCO) and simplifies the overall IT Landscape.
- **Agility and Regulatory Compliance Management:** The utility industry is rapidly evolving with new regulations and business models. C2M provides a flexible platform with advanced billing configurations and reporting tools to help utilities adapt quickly and ensure compliance with regulatory orders.
- **Back to Base and New Features:** The technical upgrade gives opportunity to assess and review the existing implementation and thereby bringing the customizations closer to back to base features available within the product. The abundance of new features available with Oracle Utilities C2M also allows to streamline and simply the complexly designed existing processes.
- **Scalability for Growth and Step towards Cloud:** The C2M is designed for utilities of all sizes and allows easy path upgrade to Oracle Utilities Cloud. It is ideal to upgrade to C2M on premise as first step (although not mandatory) to move towards smooth and swift migration to cloud.



Upgrade Approach and Execution

This section outlines the technical upgrade approach and execution steps from upgrading Oracle Utilities CCB and MDM into Oracle Utilities C2M.

Existing Application and Platform Configurations

This section provides an overview of the existing CC&B and MDM versions and the target state into which they were consolidated within C2M. Both systems contained significant customizations and integrations with multiple external applications, all of which needed to be preserved and fully supported in the upgraded C2M environment.

Oracle Utilities Customer Care and Billing	○ Oracle Utilities CCB 2.5.0.1 ○ Oracle Utilities Application Framework ○ Oracle Database 12c (BYTE Semantic) ○ Oracle Weblogic 12.1.1.3 ○ Arabic Language Pack ○ Single Sign On
Oracle Utilities Meter Data Management	○ Oracle Utilities MDM 2.3.0.2.0 ○ Oracle Utilities Application Framework ○ Oracle Database 12c (CHAR Semantic) ○ Oracle Weblogic 12.2.1.3

Oracle Utilities Customer to Meter	○ Oracle Utilities C2M 2.9.1.1 ○ Oracle Utilities Application Framework 4.5 ○ Oracle Database 19c Exadata (CHAR Semantic) ○ Oracle Weblogic 12.2.1.4 ○ Single Sign On

Technical Upgrade and Execution Steps

As part of the technical upgrade strategy, CCB and MDM were merged into the new C2M environment. The existing production instances of CCB and MDM remained untouched throughout the process to ensure business continuity. Instead, the complete upgrade lifecycle—installation, configuration, migration, and validation—was carried out on a newly provisioned and upgraded infrastructure including the application side of the product apart from the databases. This approach ensured a clean deployment path, minimized risk to current operations, and allowed for thorough testing before the final cutover.

The upgrade process is repeated multiple times in DEV, SIT, UAT and Pre-production environment with multiple mocks to ensure repeatable and efficient process has been established to meet downtime window requirements of the production cutover.

Oracle Utilities Application Upgrade

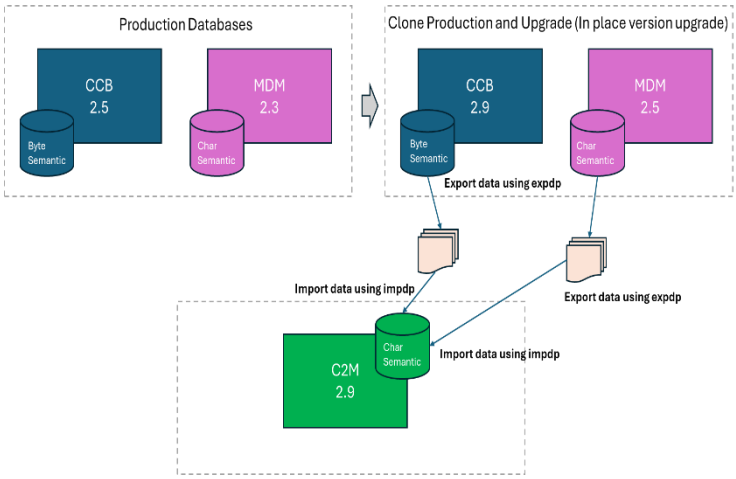
As part of upgrade strategy, recommended approach is to install latest version of the Oracle Utilities C2M (v2.9.1.1) along with required rollup patches for operation C2M Environment. The existing CCB and MDM Application instances and environments are untouched throughput the process (including lower environments). This allowed continuous checks and balances between existing CCB and MDM Environments vs. C2M Environments in terms of code and configurations.

The newly created C2M Environment is integrated with all the internal and external systems through middleware integrations. Throughput the process, the middleware integrations are untouched and ensure less/minimal impacts on overall system behavior and business processes.

Oracle Utilities Database Upgrade Approach

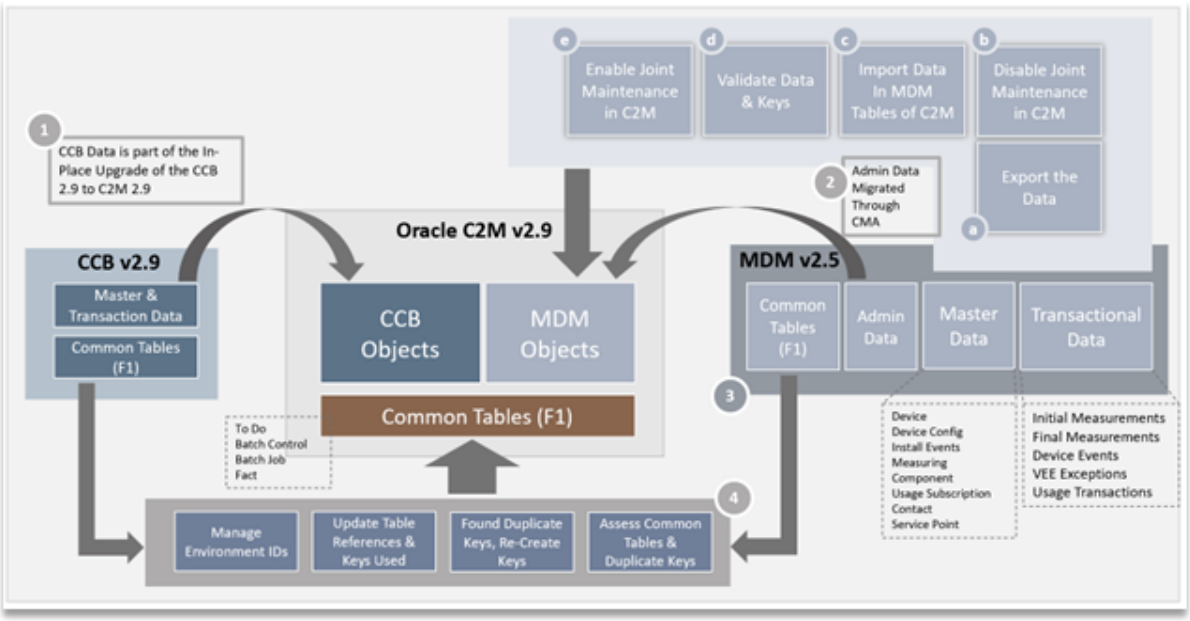
The overall complexity of the merging from CCB and MDM to C2M involved a careful planning and execution on the database upgrade. The objective is to ensure all the existing data (admin, master and transactional data) worth ~70 TB to be migrated and merged from CCB and MDM into C2M. The upgrade process involved multiple steps across both CCB, CCB and Merge process within C2M. The steps involved,

- In-place upgrade of the CCB Database
- In-place upgrade of the MDM Database
- Setup of new C2M Database with latest version / rollup
- Migration of Data from CCB and MDM using Data pump to C2M
- Handling of Common Tables during the merge process (i.e., Out Message, To Do, Sync Request etc.)



The key success factor in the entire upgrade process revolved around complexity of handling large volume of data migration from CCB and MDM to C2M and ensure all the table data from CCB and MDM are rightly merged into C2M Data model. This also involved handling large volumes of common table data between CCB and MDM and involving business and IT Stakeholders on identifying data that can be compromised due to duplicate keys.

The overall data movement is performed using the standard data pump (export and import) and is thoroughly tested for the performance and continuously fine-tuned according to the infrastructure available at hand.



Step 1 Once the in-place upgrade of CCB 2.5 to CCB 2.9 is completed, export all the data from CCB using the export data pump and import into C2M 2.9.

Step 2 All the Administrative Data will be migrated from MDM using the Configuration Migration Assistance tool available within Oracle Utilities Framework.

Step 3 All the MDM Data will be migrated from MDM v2.5 to latest version of the C2M using export and import data pump.

Step 4 The common framework tables should be analyzed one by one between existing MDM and CCB systems and will go through Data Conversion Process. Align with business and IT stakeholders on data retention for common key records.

The process is validated, tested and thoroughly reviewed in lower environments before performing multiple dress rehearsals with full size data in production size environments. This ensured trusted and repeatable data migration process steps.

The production cutover was executed in a carefully planned, phased manner to ensure minimal disruption to business operations. As part of this approach, static data—such as out-messages, initial measurements, and historical meter reads from older months—was migrated to the production environment in advance. This reduced the overall data load and allowed the team to focus on the more critical and time-sensitive information during the final cutover.

Dynamic and business-critical data, including bills, bill segments, and financial transactions, was fully migrated during the cutover window to ensure accuracy and consistency of the most recent operational data.

Optimizations and Considerations

During the upgrade process, it is important to understand the current application and database structure and look for opportunities to optimize both application and database. These optimizations will not only enhance the overall performance of application and benefit business users but will help in optimizing overall upgrade process and downtime window.

- **Database compression:** Assess compression techniques and feasibility of applying these across various tables. A mix of Advanced Compression and Hybrid Columnar Compression (HCC) is recommended based on the table size and usage.
    - ***Advanced Compression*** offers efficient row-level compression suited for frequently accessed transactional tables.
    - ***Hybrid Columnar Compression (HCC)*** delivers high compression ratios for historical and less frequently accessed data, making it ideal for archival segments and large read-only datasets.
- **Database Partitioning:** Apply partitioning to the large and volume intensive tables aligned to oracle utilities database standard guidelines and aligned to ILM Strategy. By segmenting data based on logical boundaries such as time periods or business entities, storage will become efficient, query performance improves, and maintenance operations will be drastically improved (i.e., archiving, purging etc.)
- **Database Table Archiving:** Defining and executing the Database Archiving strategies and aligning to ILM will provide greater results while during such a huge data migration and technical upgrade. Agree with customer on required vs. good to have data to define the archiving strategy (especially on transaction tables like Outbound Message, Initial Measurement, Sync Request and Response, TODOs as archiving the not needed and unused data will significantly reduce the migration data size)

Technical Upgrade Challenges and Considerations

This section outlines the typical challenges that any technical upgrade will foresee and general guidelines and best practices through which some of these can be overcome. Upgrading from legacy systems to Oracle's C2M presents several complexities related to infrastructure sizing, code retrofitting, functionality testing, cutover window planning. These issues were mitigated through meticulous planning and the implementation of appropriate tools and technologies.

Right Technical Architecture and Resources Getting right solution and technical architecture at the start of the project is very critical to success story of technical upgrade projects. It requires thorough assessment of the existing system landscape, integration architecture, business process criticalities, day in life of operations and most importantly assess the platform and applications in user in terms of data size, backup policies, upgradeability and adaptability factors from business. Any change to architecture will lead to rework and replanning of the execution – causing impacts on overall project plan schedule and overrun of budgets. On top of it, having strong technical resources who have executed and have in-depth understanding of both new and old versions is critical to project execution will ensure execution per plan.

Customization Management (CEMLI) One of the big challenges of technical upgrade is managing existing system custom configuration, extensions, java customizations and any localization changes. The challenge is multi folded when we have Customizations to be managed from both MDM and CCB, and system is continuously evolving during project life cycle. The combining and merging the Java customizations into single cm.jar should be planned carefully and thoroughly evaluated.

Release and Configuration Management Production CEMLI is continuously evolving with new demands, problems and incidents being taken care on day-to-day basis. Lack of structure around code and configuration management along with continuous retrofitting of Production code to upgrade C2M platform will lead to gaps and will lead to longer testing cycles. The amount of effort required for retrofitting should be considered into project plan by assessing the current system and reviewing the demand backlogs for current production environment.

Infrastructure Sizing and Pre-requisites Typically, defining the infrastructure sizing and pre-requisites required for technical upgrade and especially large data migration is very critical and not planning this ahead of time and as per production downtime cutover requirements would lead to last minute adjustments to process and compromise on overall cutover execution steps including window. The amount of time it will take to export and import data will largely depends on right infrastructure sizing and stability of the infrastructure during the cutover. The pre-requisites should be thoroughly reviewed, and performance needs to be tested during dress rehearsals.

Downtime Planning and Preparation The complex technical upgrade will require downtime and the amount of time required needs to be carefully planned to set the right expectations. In ideal cases, this time will largely depend on the approach to execute data movement, infrastructure provisioned and degree of parallelism through the data movement can be achieved. These factors have multiple variables which makes it difficult to assess the downtime

at start of project. Cutover planning and downtime commitments needs to be continuously evaluated and revised to ensure right expectations are set to Business Stakeholders.

Integrations and Compatibility Ensuring all the integration touchpoints are working as expected and third-party software's that are integrated with application are well intact and compatible with newly upgraded application is very critical for business process operations i.e., RPA tools, Automation Scripts and third-party reporting tools etc.

Testing and Validation is KEY Test strategy starting from Unit Testing to Business Acceptance Testing is very key to success of the technical upgrade projects involving CCB and MDM to C2M. As there will be many common processes between CCB and MDM, and business process would potentially be simplified. The need for comprehensive testing of all business processes, integrations, and reports in the upgraded environment requires substantial time and resources. New features might also need dedicated testing plans. Overlooking testing efforts and assuming that a project is mere technical upgrade will many times lead into catastrophic execution of business acceptance testing.

Document Everything and Practice More It is important to ensure delivery processes are set in place which documents each step of the upgrade process in detail – with pre-requisites, execution steps, potential risks, issues and dependencies. The key success factor in large scale technical upgrade is to arrive at repeatable and automated (as applicable) upgrade steps that will lead to successful cutover.

Sort out the Access Early It is normally overseen the importance of having right logistics during project life cycle i.e., in terms of VPN's required to access, network firewall rules and respecting the customer information security policies, network and domain configurations and access controls to execute upgrade steps. Having realized missing even a small access control during execution can cause multi-fold delays in project.

General Guidelines and Success Factors

Below are some of the general guidelines that will lead to successful implementation of the technical upgrade projects and ensure smooth post-production business day operations.

Thorough Assessment of Existing System Landscape

- Define clear system assessment strategy and plan
- Identify the complexities involved and challenges that might foresee during upgrade
- Assess customizations for the compatibility of new platform. Define impacts on external systems and communicate early.
- Categorize modules into highly impacted to no impact. This allows to track changes across these modules from production during project execution, and assess impact on technical upgrade

Global Technical Solution Design

- Start with High Level Technical Solution Design and break it down to low level execution units
- Assess Technical Complexities in terms of Data Movement, Infrastructure Needs, Pre-requisites for execution
- Set clear expectations for execution – and plan for dress rehearsals
- Optimize the current instead of optimizing the target – this will allow reduce complexity in the technical migration
- Continuously Learn and Iterate Technical Solution as and when mocks are concluded

Document and Execute

- Create detailed and tailored (to customer) technical upgrade document. Continuously evolve.
- Leverage tools to document each execution step – document execution times for each step
- Automate Pre and Post Upgrade Scripts
- List down issues faced, resolution steps and challenges seen. This will ease out cutover execution

Rigorous Testing

- Plan for multiple dress rehearsals to test the technical upgrade steps. Identify optimizations and act
- Gain experience, expose issues early, and continuously refine the technical upgrade documentation steps
- Plan multiple cycles of functional regression testing – involve business early

Cutover Plan for Everybody

- Cutover Plan should be owned by everyone – define responsibilities and ownership
- Involve Stakeholders Early
- Define the cutover window and plan technical activities around. Define a realistic window based on the system assessment, infrastructure, network bandwidth availability and dependencies with respect to external window. Plan for business continuity.