

Case Study

Re-architecting Applications to Derive Best Value from Oracle Communications Suite



The Customer

Miami-Dade Water and Sewer Department (MDWASD) provides drinking water and wastewater services, promotes water conservation and works to safeguard public health and the environment. Miami-Dade County maintains more than 8,500 miles of underground water lines, as well as approximately 4,100 miles of sewer lines, serving more than 2.3 million residents and thousands of visitors.

The multi-year Capital Improvement Program ensures that high quality drinking water and wastewater services are in compliance with all regulatory requirements. Water conservation efforts help protect this important natural resource while also educating future generations on the importance of smart water use.

The Challenge

- In the process of designing new features, MDWASD found themselves needing a common architecture as a template for new features
- MDWASD needed this new feature solution to have the following capabilities:
 - Centralized business logic within Oracle's CCB (Customer Care and Billing) framework.
 - Cross reference this process with existing CCB objects
 - Self-contained entity with its own lifecycle that can be further exposed for web service used.

About Cogent Infotech

Founded in 2003, Cogent Infotech is a trusted, award-winning firm with **21+ years** of experience, **150+** government contracts, **10,000+** projects, and a 96% employee retention rate. Recognized as an SBA Small Business and MBE-certified, we deliver excellence through diverse talent, AI-driven recruitment, and cooperative contracts like NASPO Value Point and TIPS-USA.



Contact Us
+ 972-200-0109



www.cogentinfo.com



Pittsburgh, PA, USA
1035 Boyce Rd,
Pittsburgh, PA 15241

The Process & Solution

MDWASD was presented with options:

- Customized Framework Entity – MDWASD can create their own custom entity. There were 2 ways to do this:

1. Java-Metadata Custom Entity

- This is how custom objects were created prior to Configuration Tools.
- This was done by creating custom tables, and metadata, and then writing maintenance programs using Java.
- This was also previously done using COBOL.
 - **PROS** – Relatively simple to design and implement, team needed no additional training.
 - **CONS** – Upgrade issues (possible deprecation), requires builds, deployments, and downtime.

2. Business Objects

- This is a more recent implementation, using Configuration tools.
- Configuration Tools is a more front-end-leaning way to develop customizations, aimed towards business users.
- All business logic is created via configuration instead of compiled java code.
 - **PROS** – Upgrade-proof, greatly flexible, no downtime required.
 - **CONS** – Complex implementation, very high learning curve.

The Impact

- MDWASD Opted to use Cases: Deciding factors were a CCB upgrade soon (meaning deprecation was a problem) and learning curve meant time to train new developers.
- Use Case Functionality helped in the following:
 - Oracle CCB created simpler objects with a more rigid structure than custom entities
 - Cases were meant to be used to encapsulate business processes and give them “physical” form in the framework.
 - Cases allowed the creation of objects that followed a configurable lifecycle, via states and plugin points for custom code within these states
- MDWASD already had a few developers being trained with Cogent-led Case Design and Development since the inception of the project.

