



The Critical Next Step for Interoperability:  
Designing and Implementing Interfaces  
between Standards

Gary McNaughton, MultiSpeak Initiative  
Grid-Interop 2012

# Agenda

- Mapping Between Standards: What and Why
- Overview of MultiSpeak and OpenADR
- Mapping Methodology
  - Common business processes
  - Actors and domains
  - Use Cases and Sequence Diagrams
  - Gap analysis
- Results for MultiSpeak to OpenADR
- Conclusions

# What and Why of Mapping

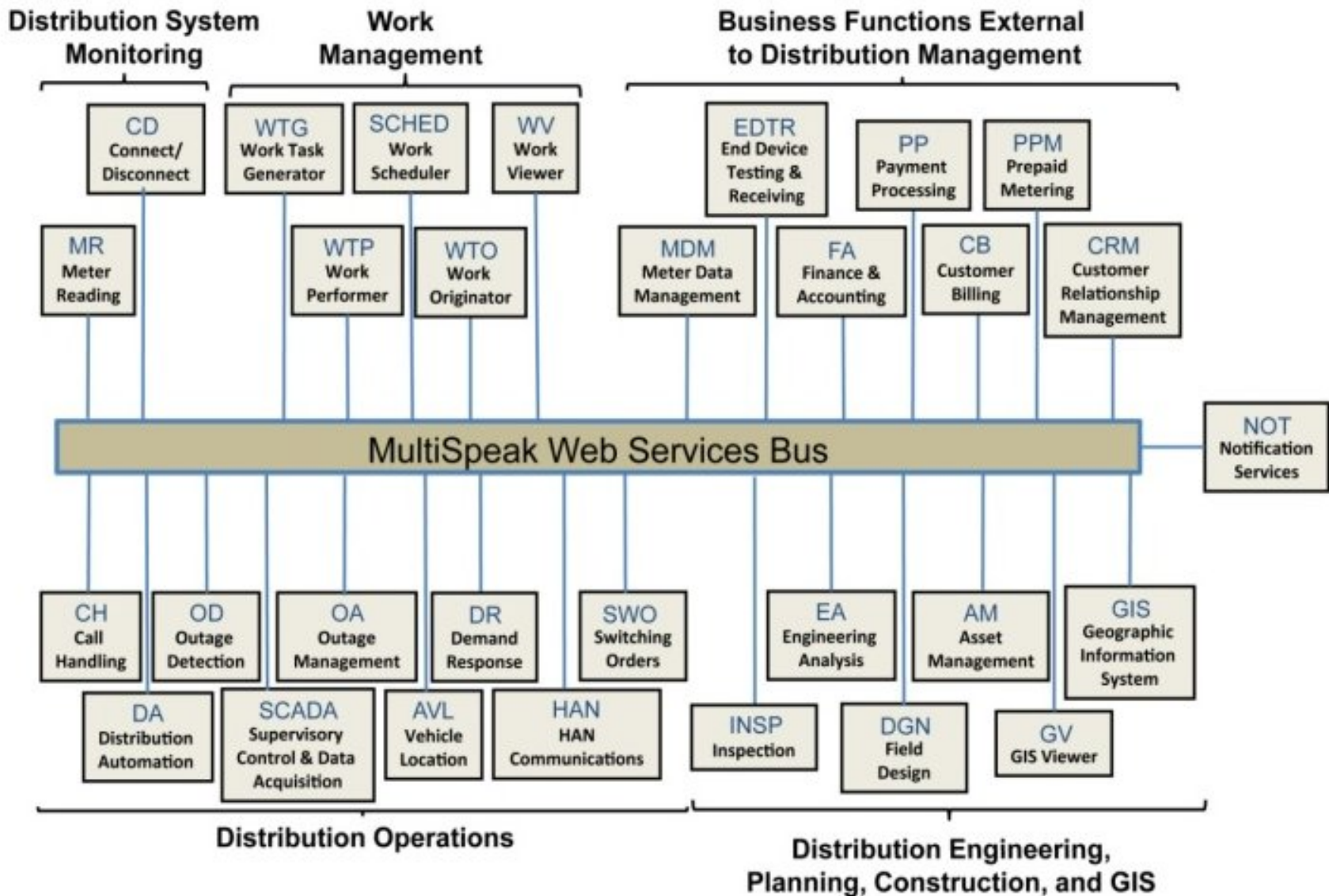
- In the real world, products are deployed in a system: different components based on different standards
  - For example, MultiSpeak initiates DR event implemented by an OpenADR system
- How are inter-standard interfaces evaluated (mapped) to ensure standardized inter-standard interoperation?
- What is an efficient methodology for mapping?



*Driving to Grid 2020*

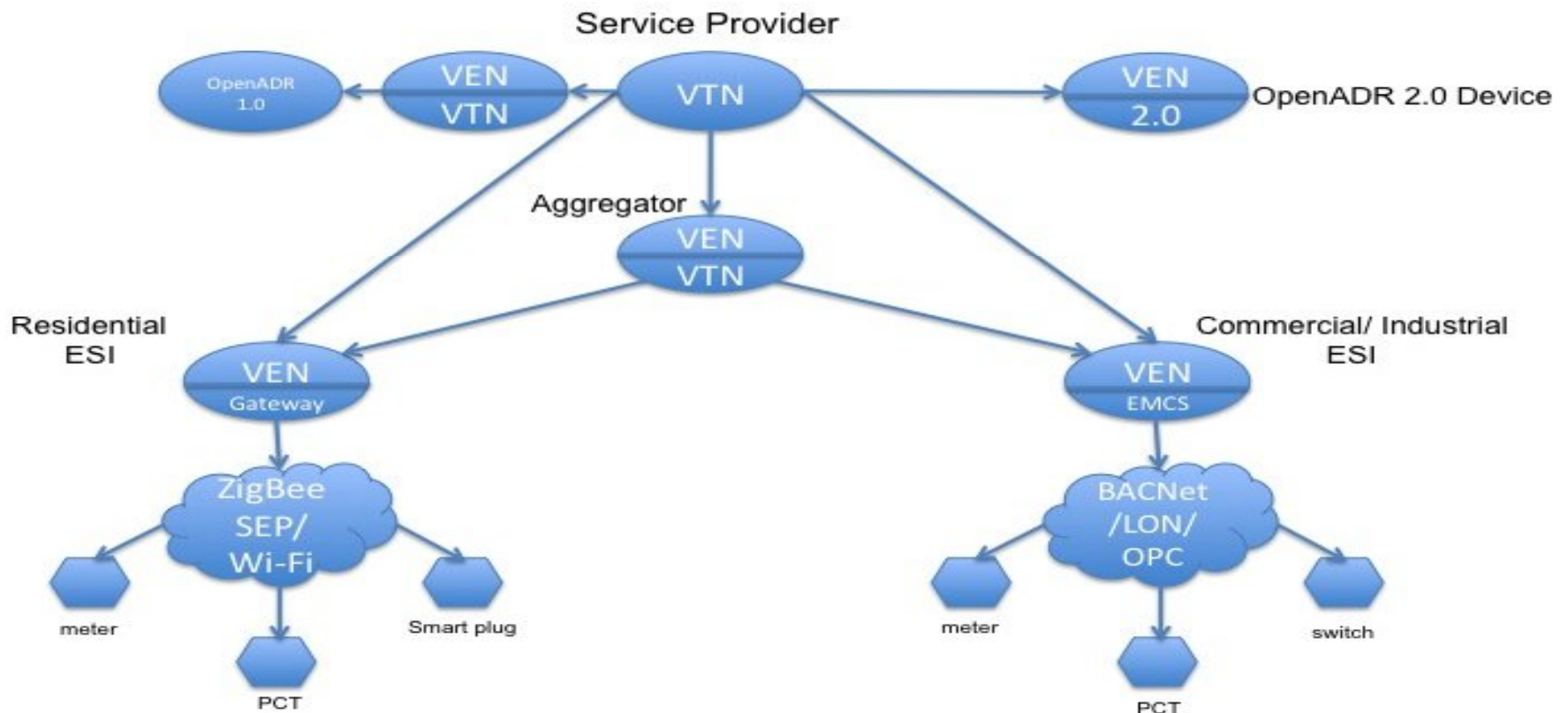
# OVERVIEW OF MULTISPEAK AND OPENADR

# MultiSpeak Version 4.1.5



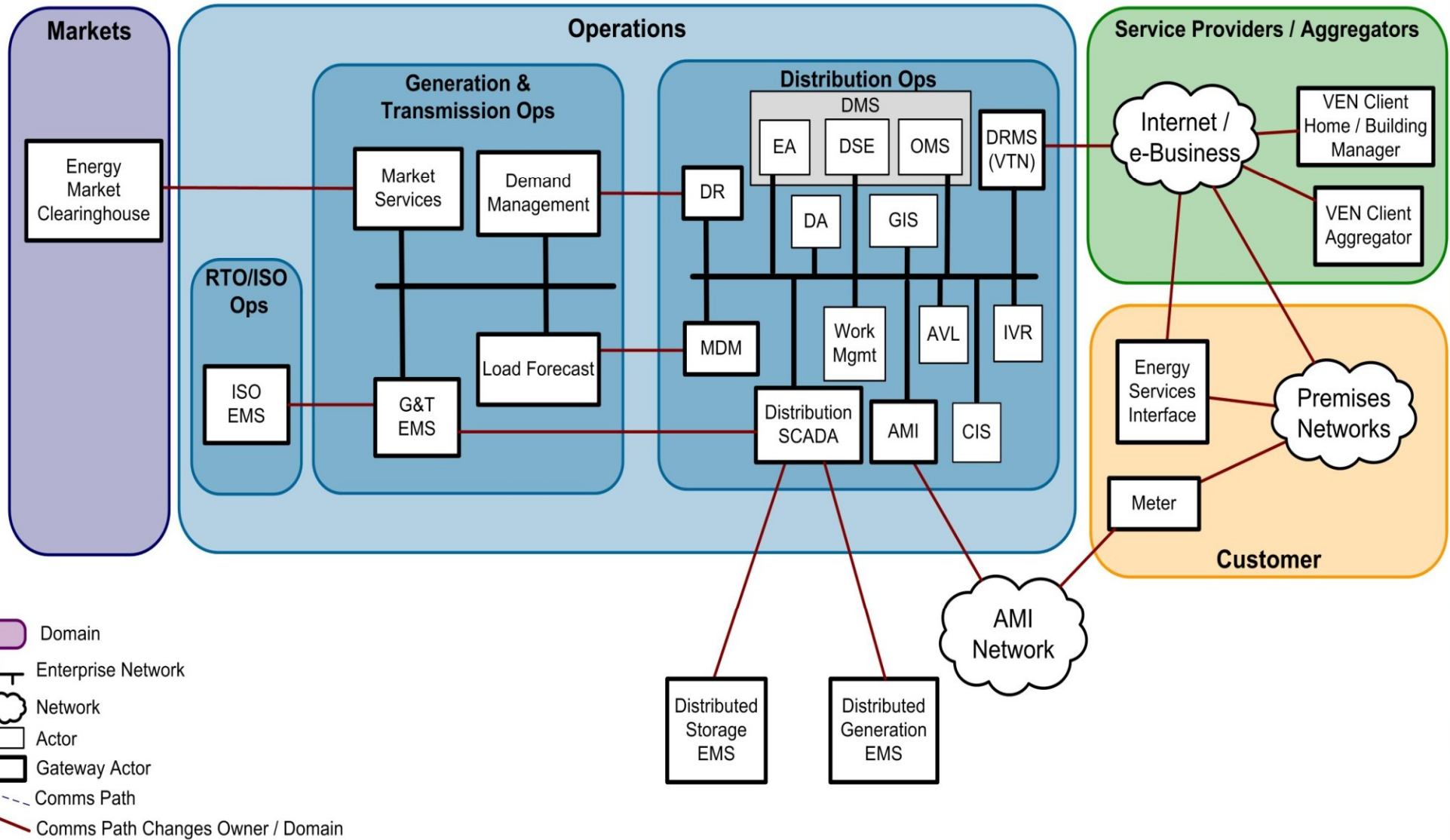
# OpenADR 2

## Conceptual OpenADR 2.0 Interoperability Demonstration





# MultiSpeak Domain and Utility Interfaces





*Driving to Grid 2020*

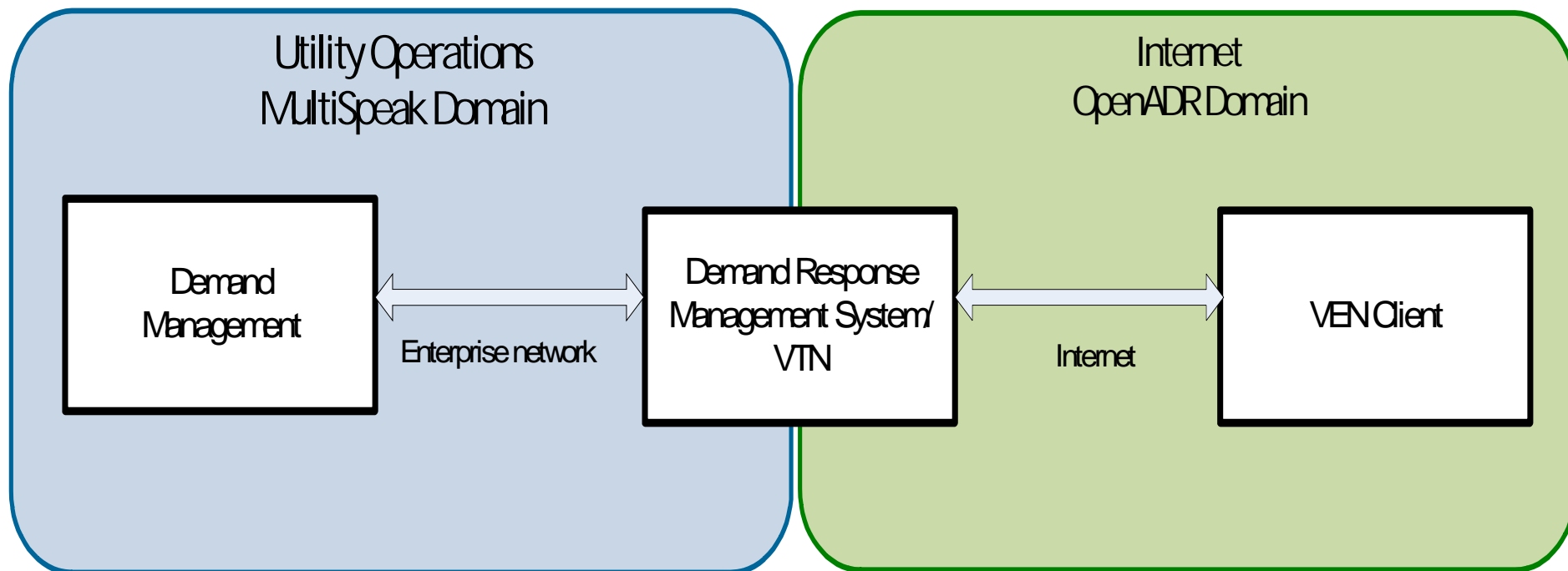
# MAPPING METHODOLOGY



# Business Process

- Major business processes
  - The utility manages its demand response resources by distributing events to customers with responsive assets. Customers may choose to participate in one or more events.
  - The utility distributes/updates critical peak price events to customers enrolled in a program. Customers may choose to participate in one or more events.

# MultiSpeak to OpenADR



# Business Process (Continued)

- MultiSpeak domain, the actor is a demand management application
- Actor in the OpenADR domain: Virtual End Node (VEN) interface to the utility
- The bridge between the two domains: a third actor (an adaptor application): the Demand Response Management System (DRMS).

# DR Use Cases

Utility manages demand response event to customer demand responsive resource(s).

- MSP\_OAD:1 Utility issues demand response event to customer demand response resource(s) (PUSH Method)
- MSP\_OAD:2 Utility cancels active or future demand response event (PUSH Method)
- MSP\_OAD:3 Utility modifies demand response event (PUSH Method)
- MSP\_OAD:7 VEN requests list of active events (PULL model)

# CPP Use Cases

The utility manages price signals to customers, who decide how to respond using their demand resources.

- MSP\_OAD:4 Utility issues critical peak price event (PUSH Method)
- MSP\_OAD:5 Utility cancels active or future critical peak price event (PUSH Method)
- MSP\_OAD:6 Utility modifies critical peak price event (PUSH Method)
- MSP\_OAD:7 VEN requests list of active events (PULL model)

Utility issues demand response event to customer demand response resource(s) (PUSH Method)

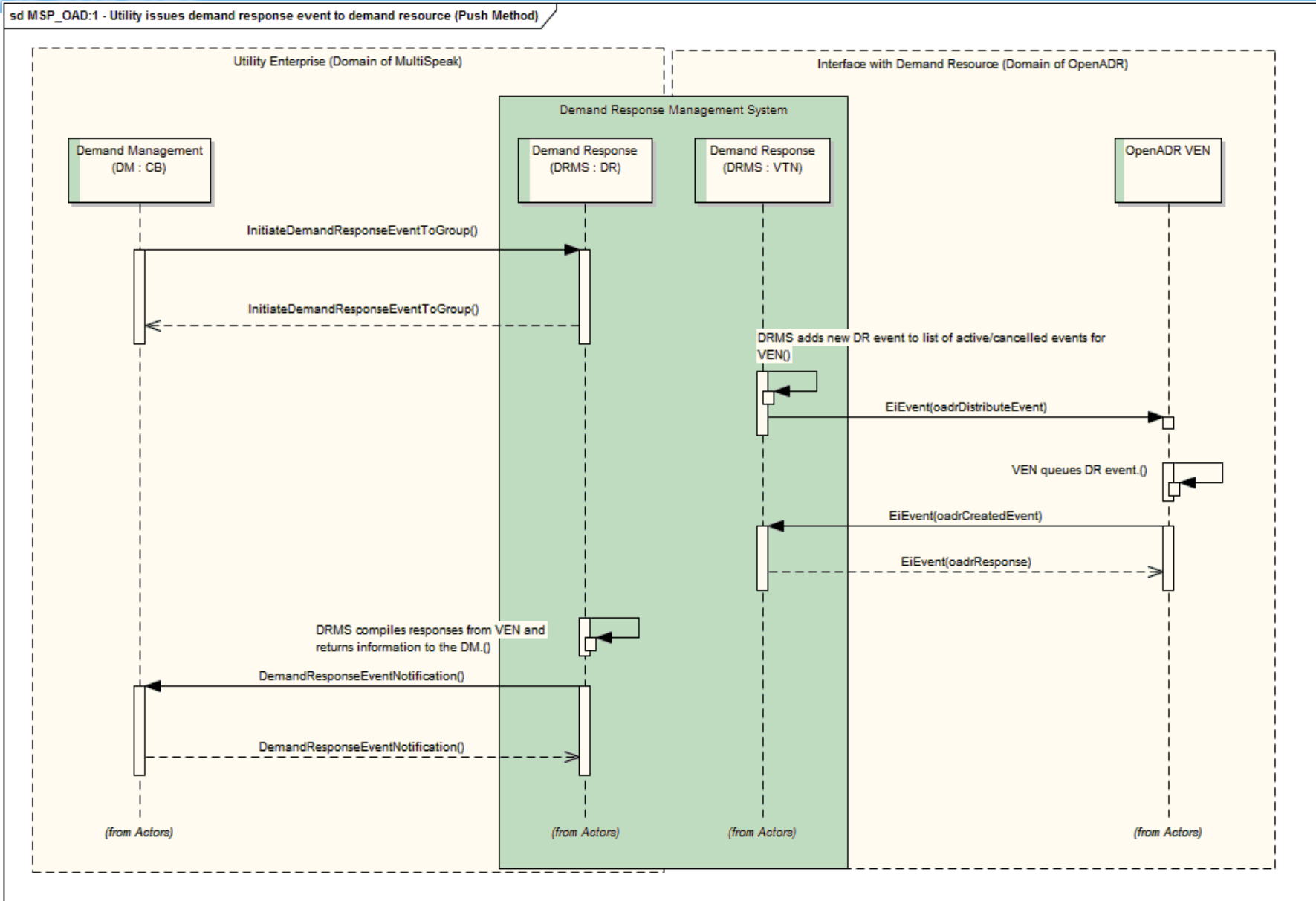
- MultiSpeak Message:
  - InitiateDemandResponseEvent
  - InitiateDemandResponseEventToGroup
  - DemandResponseEventNotification
- MultiSpeak Object used in the Message Payload:
  - demandResponseEvent
  - demandResponseEventStatus
- OpenADR Service:
  - EiEvent
- OpenADR Payload:
  - oadrDistributeEvent
  - oadrCreatedEvent

# Data Element Mapping

- Mapping from one standard to another is not always direct. The following mapping terms were defined:
  - Direct: A mapping can be done directly from one schema element to the other.
  - Derived: The value or content of the destination element can be derived from other information, such as the function name, other elements, etc. Examples of recommended values or algorithms may be provided.
  - NA: The element being mapped is specific to the standard or does not apply to the specific use case.
  - Computed: The data element can be computed (elements such as timestamps would fall under this category).
  - UsageGap: Required to implement the functionality in the standard that is being mapped if it is not directly available and cannot be computed or derived.
  - ExtGap: The standard being mapped has additional features that are not currently supported by MultiSpeak usage models or data objects and cannot be computed or derived.



# Example Sequence Diagram



# Example Mapping Table

MultiSpeak Methods: Initiate/Cancel Message Object: demandResponseEvent		Mapping	OpenADR 2.0a Profile Object: oadrDistributeEvent.oadrEvent.eiEvent.eiActivePeriod	
Element	Description		Element	Description
<eventStartTime>	Start time for the DR event, if not specified, then the start time is now	Direct	<properties.dtstart>	Start time of the event
<eventDuration>	Duration of the DR event, if not specified, then the duration is forever	Direct	<properties.duration>	Duration of the event, if the attribute is 0, then the event goes forever
		NA	<properties.tolerance.startbefore>	This allows the definition of a random start time, and is ignored by VENs in 2.0a
<randomizeEventStart>	Apply a randomized dither to the start of an event (a Boolean value)	Derived: use a default value if the element is set to "true"	<properties.tolerance.startafter>	This allows definition of a random start time after the beginning of an event
		Derived: Default to 0	<properties.x-einotification.duration>	Length of time for notification of the event (possibly no functional effect in 2.0a)
		ExtGap2	<properties.x-eiRampUp>	Ramp up period and is used in 2.0a profile to determine when the event status transitions from "far" to "near"
		ExtGap2	<properties.x-eiRecovery>	Event recovery period
		NA	<components>	Placeholder to maintain schema conformance with Energy Interop



*Driving to Grid 2020*

# RESULTS



*Driving to Grid 2020*

# Results

- Basic difference in assumptions about capability of receiving entity, results in minor usage gaps.
- The functions, methods and data objects contained in MultiSpeak Version 4.1.5 are sufficient to send demand response and critical peak price events to an interface that implements the OpenADR 2.0a profile.
- No critical usage gaps (gaps where a particular OpenADR functionality cannot be supported by MultiSpeak) were identified.
- Two minor usage gaps and several extension gaps (capabilities in OpenADR that would enrich MultiSpeak if introduced into the MultiSpeak specification) were noted.

# Mapping Gaps

- OpenADR supports the ability to “modify” events. MultiSpeak currently only supports the verbs of “Initiate” and “Cancel” for the operations on the data objects. A flow was described that allows for modification of events using the capabilities present in MultiSpeak Version 4.1.5.
- OpenADR event signals support more than one interval. The current definition of DR and critical peak price events in MultiSpeak are limited to one interval and therefore one value over the duration of the event.
- OpenADR eiEvent includes attributes that define notification duration; ramp-up and recovery times for the event for examples as to how they relate to the event. These periods do not exist in the MultiSpeak event objects.
- A critical peak price event by definition implies that the price for the event is at the highest level. The EiEvent <signalPayload> attribute allows for events to have 4 values (normal, moderate, high or special). The event objects in MultiSpeak do not provide for representing other values for price besides “peak”



*Driving to Grid 2020*

# CONCLUSIONS

# Conclusions

- The mapping of functions between the MultiSpeak and OpenADR 2 standards are analyzed and presented using a use case methodology based on business processes.
- The results:
  - Comprehensive description of how MultiSpeak maps to OpenADR 2
  - Quick reference and guide for developers and system implementers.
  - The methodology developed for this project can serve as a model for other mapping efforts.