



---

# Analysis of Bridges & Wormholes Paper

Part 1 – How Control Works

First Delivered: November 24, 2025

Last Updated: December 8, 2025

Michael M Lee



## Purpose of Analysis Effort

- Understand the wormhole approach to bridging
- Assess its viability
- Provide an overview of the approach that makes it more accessible
- Identify possible improvements

## Key Terms

- Home (requesting service) domain
- Away (providing service) domain
- MX = Model Execution/Software Architecture domain

# Major Concepts in B&W Paper



## Pre-existing Concepts (Bridge Services in Away)

- External Event
  - Synchronous Service
- 
- Control Reception Point (CRP)

## New Concepts

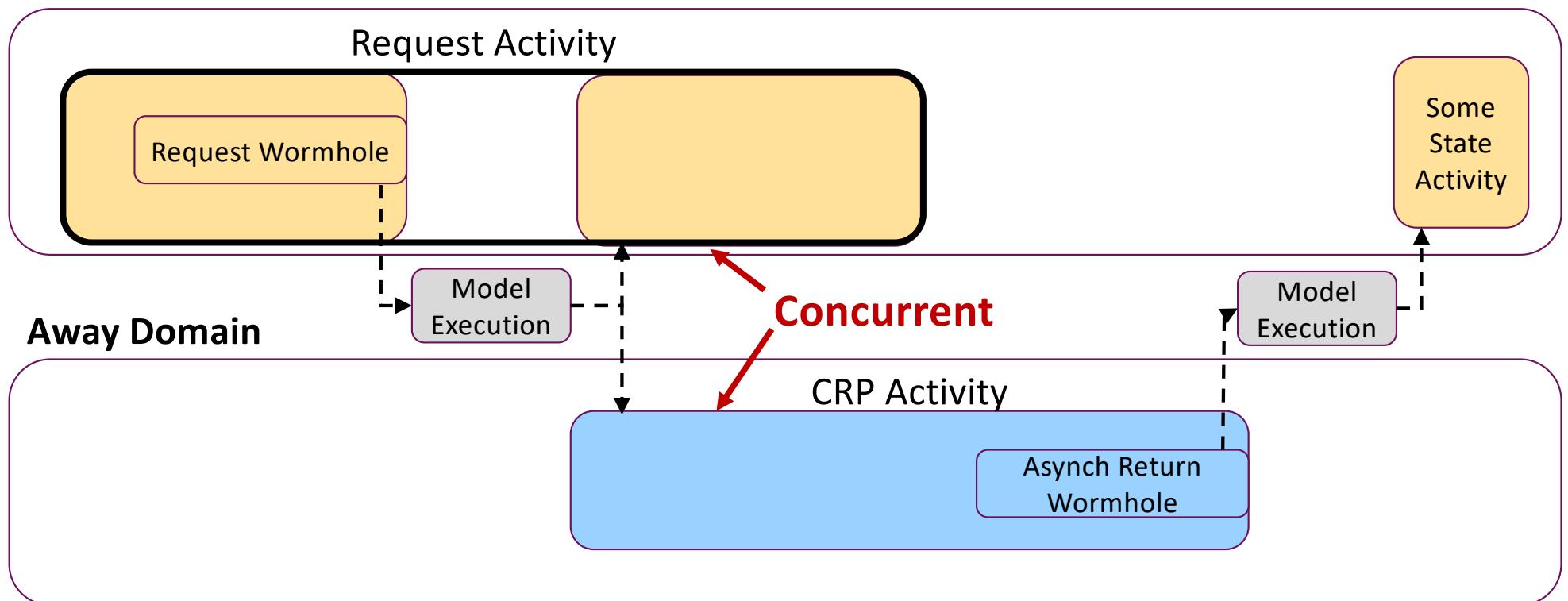
- **Request Wormhole** – New action in Home to request a bridge service
- **Synchronous Return Wormhole** – New action in Away to effect a synchronous return
- **Return Coordinate** – Information required to perform a synchronous return. If needed, created by MX, passed via MX, stored in Away, used in Away with Synchronous Return.
- **Asynchronous Return Wormhole** – New action in Away to effect an asynchronous return
- **Transfer Vector** – Information required to perform an asynchronous return. Started in Home, completed by MX, stored in Away, used in Away with Asynchronous Return.

# Asynchronous Behavior and Concurrency

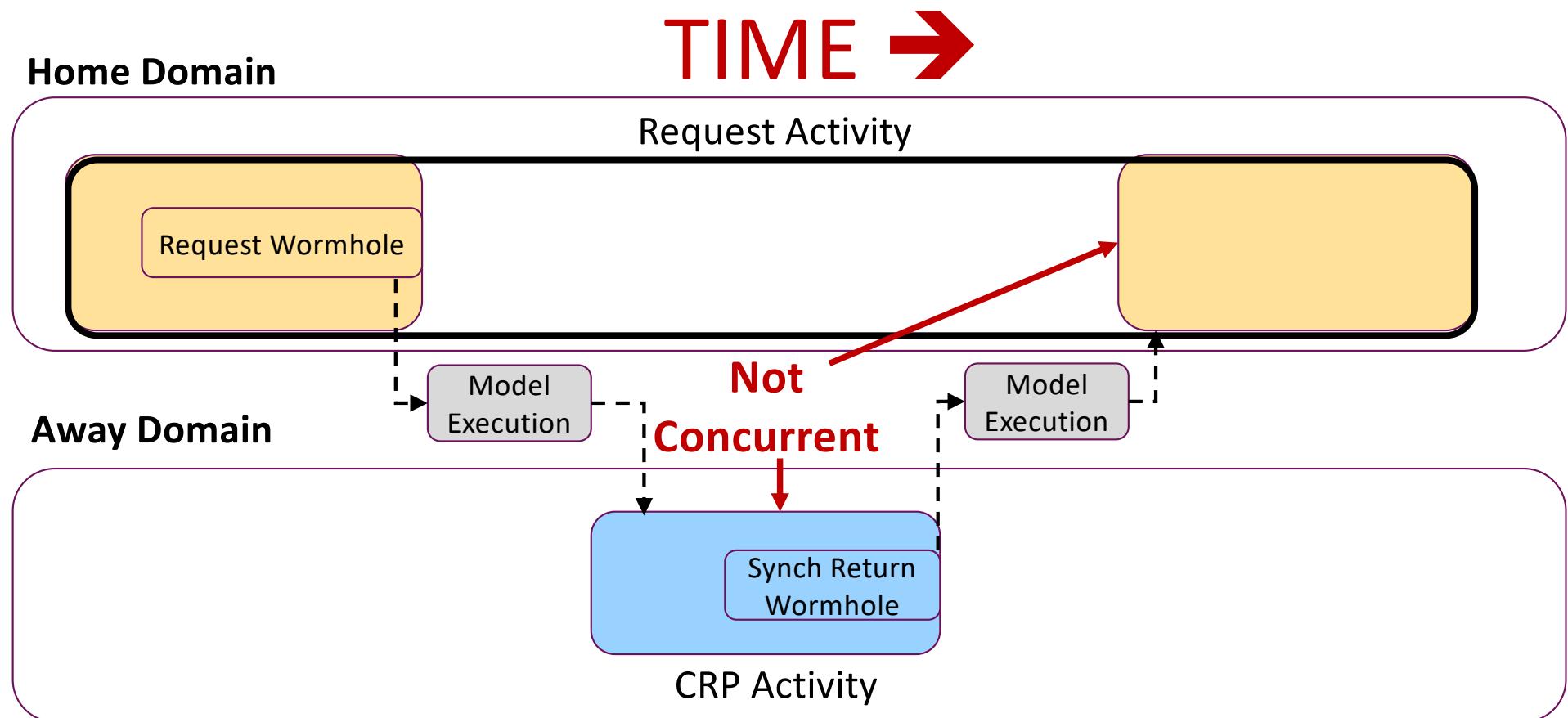


Home Domain

TIME →



# Synchronous Behavior and Concurrency





# Initial Analysis Focus

## Bridge Development Phases

- **Specification:** Specify Bridge Endpoints
- **Connection:** Connect Bridge Endpoints
- **Execution:** Develop MX code to support connected Bridge Endpoints

## Additional Details

- **Specification:** Specify Bridge Endpoints
  - Request Wormholes, External Events, Synchronous Services and Return Wormholes
    - Behavior 
    - Data
- **Connection:** Connect Bridge Endpoints
  - Match Request Wormholes to External Events or Synchronous Services 
  - Populate the participating domains
  - Develop correspondences between data values that cross the domain boundary and address semantic shifts between domains
- **Execution:** Develop MX code to support connected Bridge Endpoints

# Synchronous Service vs External Event (CRPs)



## Reminder

- An activity that provides a bridge service is activated in the Away domain:
  - by an event (External Event) because the activity to provide the service needs to be coordinated with other activities possibly executing in the Away domain or
  - by an invocation (Synchronous Service) because the activity to provide the service does not need to be coordinated with other activities possibly executing in the Away domain

## Implications

- This difference says nothing about whether the Home domain will experience the service asynchronously or synchronously. The Home domain has to indicate this via the Request Wormhole.
- Both CRPs are simply an activity, one activated by an event and one by an invocation. They have identical capabilities and behavior.
- Both CRPs can contain a synchronous return or an asynchronous return or neither.

# Specification for Control Reception Point Interface (Away)



## Interface Specifications



- ID
- Meaning
- Processing specified in CRP activity
- Inputs (to activity)
  - Data items (pass by value)
  - Return Type (stored by activity if present)
    - Transfer Vector (TV) or
    - Return Coordinate (RC)\* or
    - None
- Outputs (requires a TV or RC input)
  - Data Items (pass by value)
  - Stored Return Type

\*Produced by software architecture if required

## Types of Control Reception Points

Type	Return Type	Outputs
CRP-1	Return Coordinates	Y
	Return Coordinates	N**
CRP-2	Transfer Vector	Y
CRP-3	Transfer Vector	N
	None	Y***
CRP-4	None	N

\*\*Not supported. Return Coordinate not produced w/o Outputs. (Case #1, pg 5)

\*\*\*Not supported. Outputs require a Return Coordinate or a Transfer Vector

# Behavior for a Control Reception Point (Away)

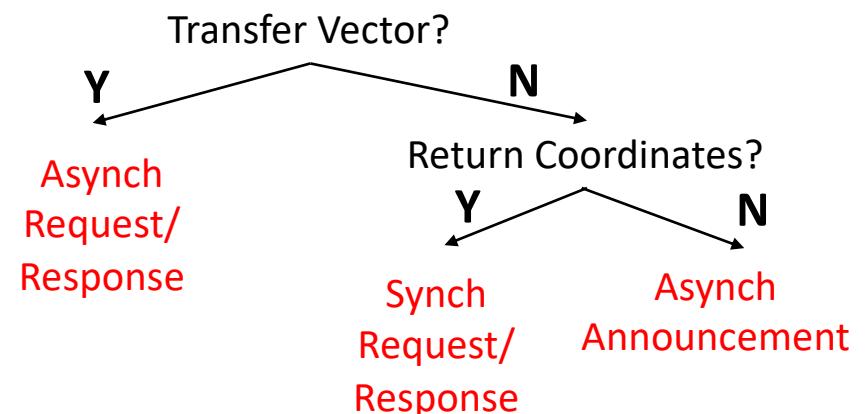


## Types of Control Reception Points

Type	Return Type	Outputs
CRP-1	Return Coordinates	Y
CRP-2	Transfer Vector	Y
CRP-3	Transfer Vector	N
CRP-4	None	N



## Behavior Provided By Control Reception Points



## Behavior of Control Reception Points

Type	Behavior
CRP-1	Synch Request/Response
CRP-2,3	Asynch Request/Response
CRP-4	Announcement



# Specification for Request Wormhole (Home)



## Specifications

- ID
- Meaning
- Input data items
- Output data items
- Transfer Vector or None



## Types of Request Wormholes

Type	Return Type	Outputs
RQW-1	No Transfer Vector	Y
RQW-2	Transfer Vector	N
RQW-3	Transfer Vector	Y
RQW-4	No Transfer Vector	N

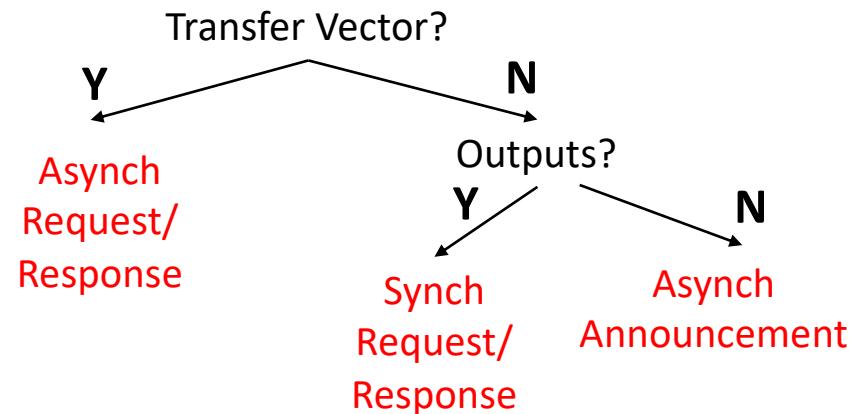
## Behavior for Request Wormhole (Home)

### Types of Request Wormholes

Type	Return Type	Outputs
RQW-1	No Transfer Vector	Y
RQW-2	Transfer Vector	N
RQW-3	Transfer Vector	Y
RQW-4	No Transfer Vector	N



### Behavior Expected By Request Wormholes



### Behavior of Request Wormholes

Type	Behavior
RQW-1	Synch Request/Response
RQW-2,3	Asynch Request/Response
RQW-4	Announcement



# Matching a Request Wormhole with a CRP via Spec



## Types of Request Wormholes

Type	Return Type	Outputs
RQW-1	No Transfer Vector	Y
RQW-2	Transfer Vector	N
RQW-3	Transfer Vector	Y
RQW-4	No Transfer Vector	N

## Types of Control Reception Points

Type	Return Type	Outputs
CRP-1	Return Coordinates	Y
CRP-2	Transfer Vector	Y
CRP-3	Transfer Vector	N
CRP-4	None	N

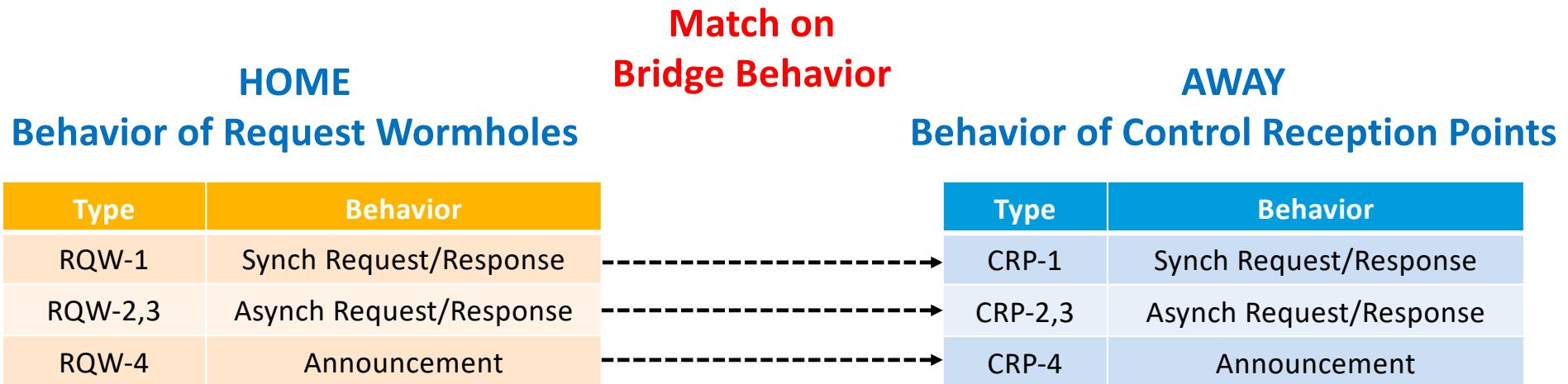


## Constructing the Bridge:

“For each request wormhole in one domain (Home), we must identify a synchronous service or an external event (CRP) in another domain (Away) .”

- Bridges & Wormholes paper

# Matching a Request Wormhole with a CRP via Behavior



## Observations

1. Typing the behavior of Request Wormholes and CRPs facilitates matching.
2. Matching I/O data items remains to be done for final match.
3. A given CRP (bridge service) in Away can only support 1 type of bridge behavior based on the type of return wormhole it contains.
4. Thus the behavior of a bridge is determined by the Away domain not the Home domain.

# Specifying Return Wormholes (Away)



## Synchronous Return Wormhole

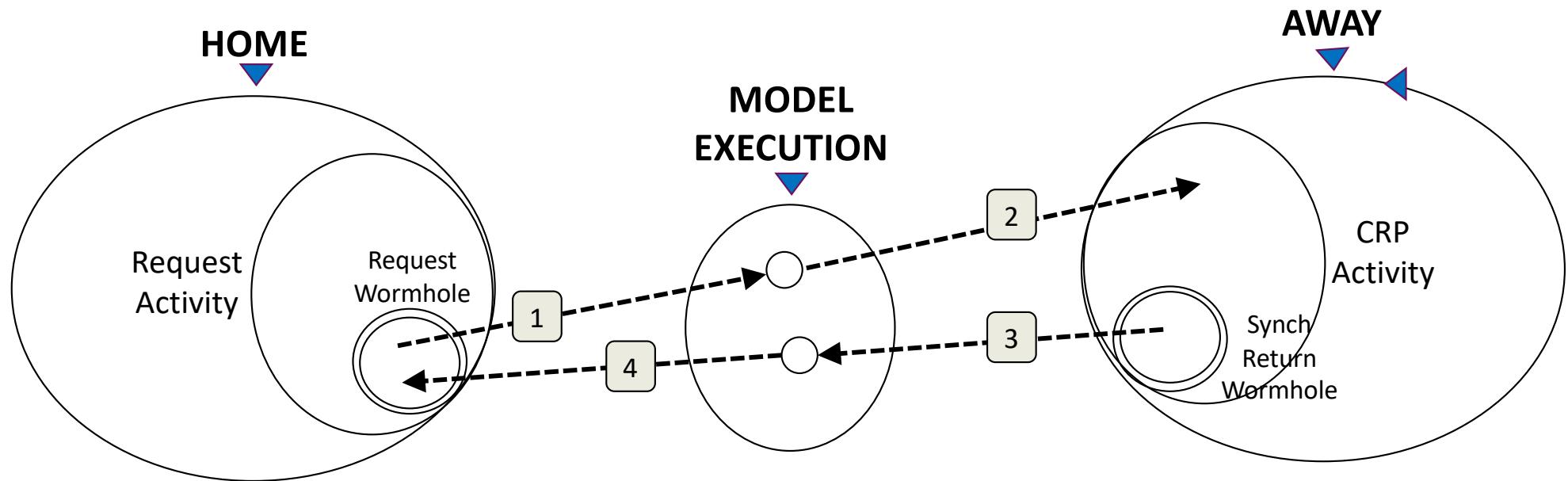
- ID
- Meaning
- Return Coordinate
- 1+ “return” data items

## Asynchronous Return Wormhole

- ID
- Meaning
- Transfer Vector
- $0 \rightarrow 1+$  “return” data items

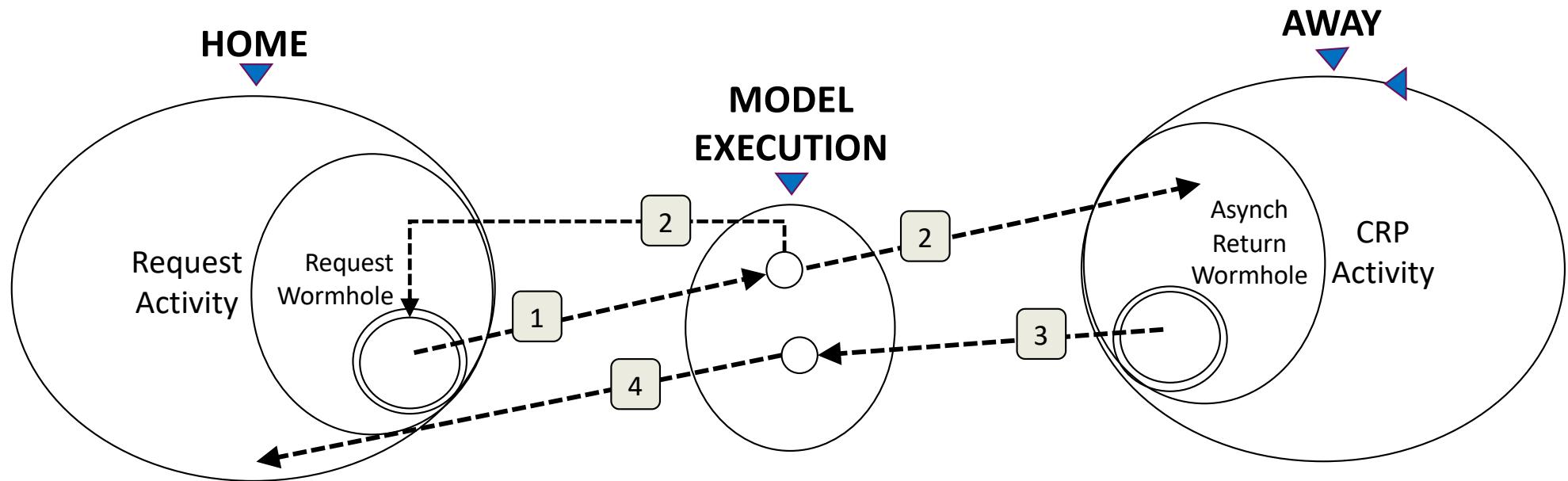
**Remember:** Announcement behavior does not require a Return Wormhole

# Simplified View of a Synchronous Bridge Scenario



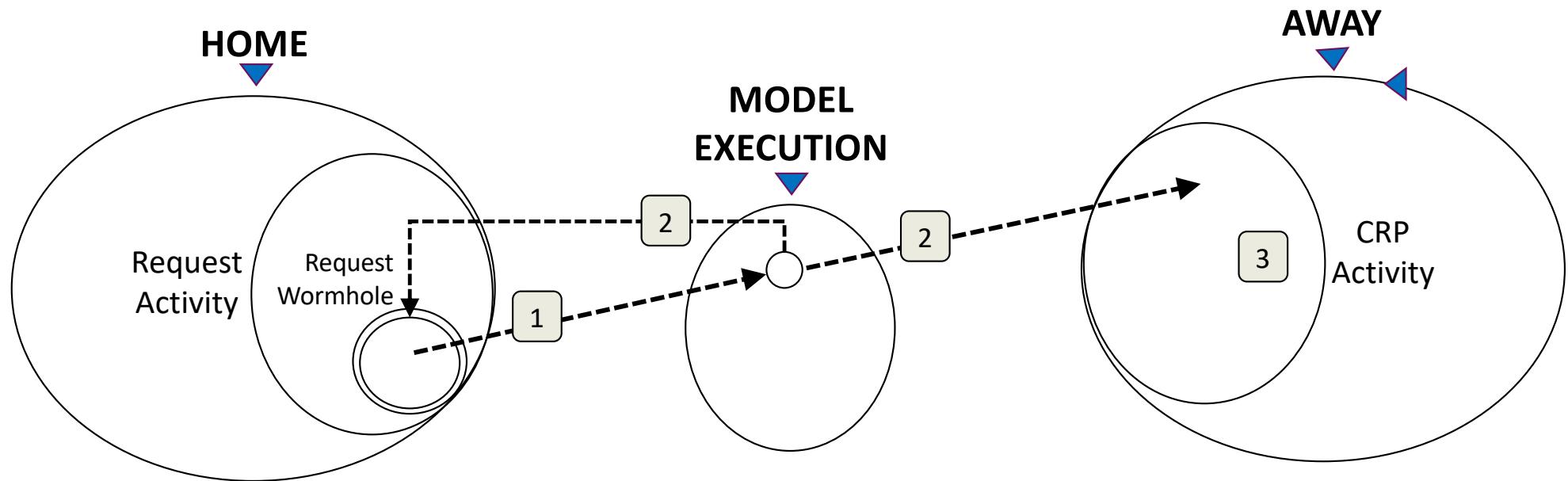
1. A Request Wormhole action executes in **Home** which suspends the Request Wormhole action and causes MX to...
2. Initiate activation of the associated CPR activity (w/input data ) in **Away** which executes until...
3. Eventually, a Synchronous Return Wormhole action executes in the CRP Activity in **Away** which causes MX to...
4. Resume the Request Wormhole action in **Home** (w/requested return data).

# Simplified View of an Asynchronous Bridge Scenario



1. A Request Wormhole action executes in **Home** which suspends the Request Wormhole action and causes MX to...
2. Initiate activation of the associated CPR activity in **Away** and resume the Request Wormhole action in **Home**.
3. Eventually, an Asynchronous Return Wormhole action executes in the CRP Activity in **Away** which causes MX to...
4. Post the “Populated Return Vector” event in **Home**

# Simplified View of an Announcement Bridge Scenario



1. A Request Wormhole action executes in **Home** which suspends the Request Wormhole action and causes MX to...
2. Initiate activation of the associated CPR activity in **Away** and resume the Request Wormhole action in **Home**.
3. Eventually, the CRP Activity in **Away** completes without executing any return wormhole actions.

Questions? Comments?? Suggestions???

---



---



# Preview of Issues and Proposed Improvements

## As-Is Bridges & Wormholes

- 1 Request Wormhole to address:
  - synchronous,
  - asynchronous and
  - announcement behavior
- 2 Return Wormholes to address:
  - synchronous and
  - asynchronous behavior

## Results

- CRP can only provide 1 behavior based on Return Wormhole used
- Away determines bridge behavior
- Home must match Away behavior

## Improved Bridges and Wormholes

- New data type, Return Mechanism w/subtypes:
  - Transfer Vector
  - Return Coordinates
- 3 Request Wormholes, 1 per behavior
  - Explicit interface
  - Clarifies Home's desired behavior
- 1 Return Wormhole
  - Away provides stored Return Mechanism
  - MX uses subtype to provide requested behavior

## Results

- CRP can provide both synch and asynch behavior
- CRP can provide different behavior to different Homes
- Home determines bridge behavior