Comparing UML-RT and xtUML

Outline

Assuming 2016 state-of-the-art and planning

- Methodology
- Notation
- Workflow
- Tool Architecture
- Convergence Possibilities
- Roadmap

Methodology

UML-RT (ROOM)	xtUML (Shlaer-Mellor)
elaborative	translative
functional decomposition	domain based
encapsulation	relational
executable	executable
abstract	more abstract
architecture recognized	architecture domain (MC)
fixed run-time	run-time rules

Notation

UML-RT	xtUML
Capsule	Component
Protocol	Interface
C++ Class	Class
	Association
Harel State Chart	State Machine
C++	Action Language

Workflow

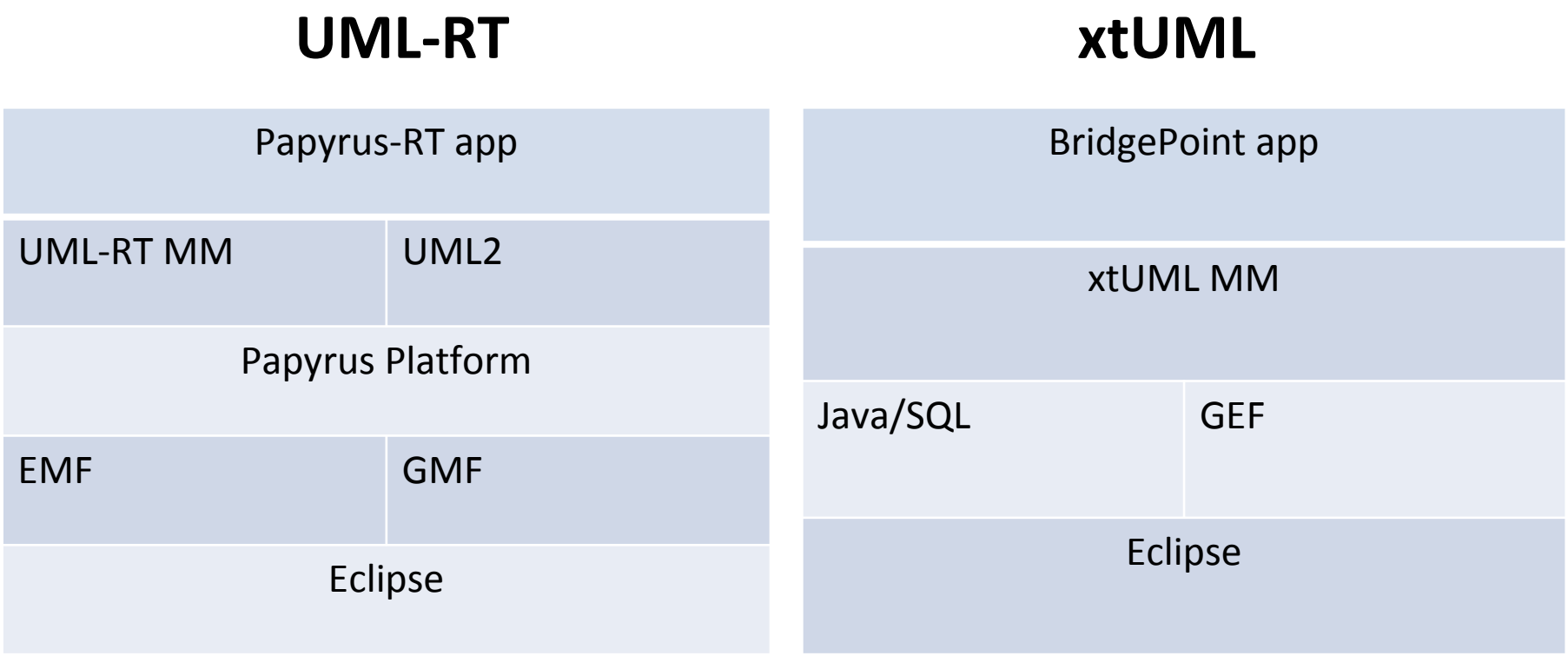
UML-RT

1) top capsule
2) decomposition
3) messaging/protocol
4) state behavior
5) progressive refinement
6) activities

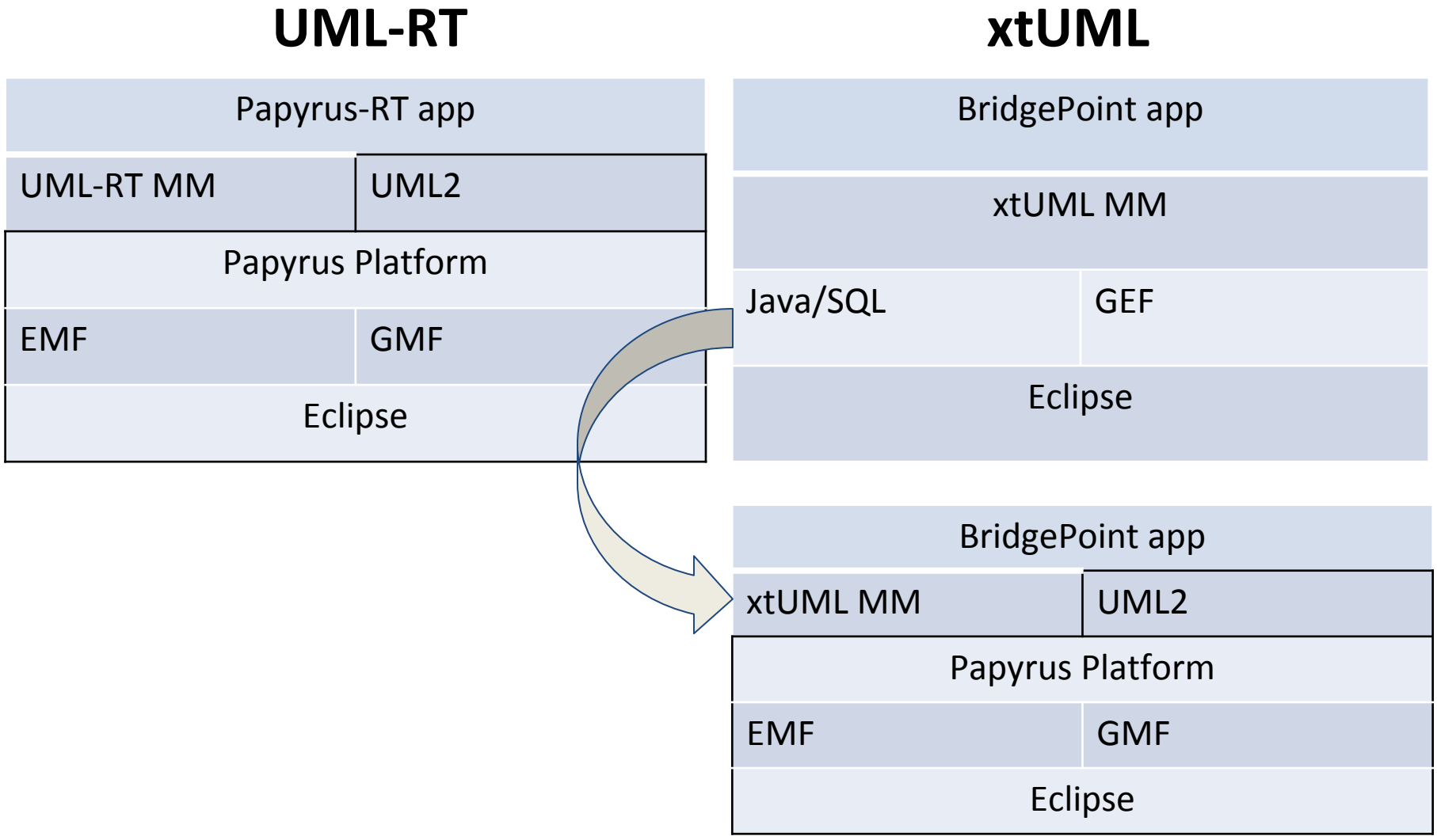
xtUML

1) domain separation
2) classes, associations and attributes
3) state behavior
4) activities
5) component deployment

Tool Architecture



Tool Architecture



Possibilities for Convergence

- merged methodology
- combined notation
- integrated workflow
- single tool
- common tool architecture

Possibilities for Convergence

- ~~merged methodology~~
  - "major differences" in data modeling, control and actions
- ~~combined notation~~
  - complex, confusing and overlaps with different meanings
- ~~integrated workflow~~
  - simpler is better and WoW are not the same
- ~~single tool~~
  - single architecture but unique dialect
- common tool architecture

Roadmap

1. Package BridgePoint as Papyrus-xtUML now.
   - alongside Papyrus-RT, Papyrus-SysML, etc.
2. Converge to common tool architecture.
   - EMF, ~~SQL~~, UML2, Papyrus Platform
3. Consider additional commonality as time progresses.
4. Establish clean, hybrid interoperability of UML-RT and xtUML systems.