Messaging

The unloved paradigm

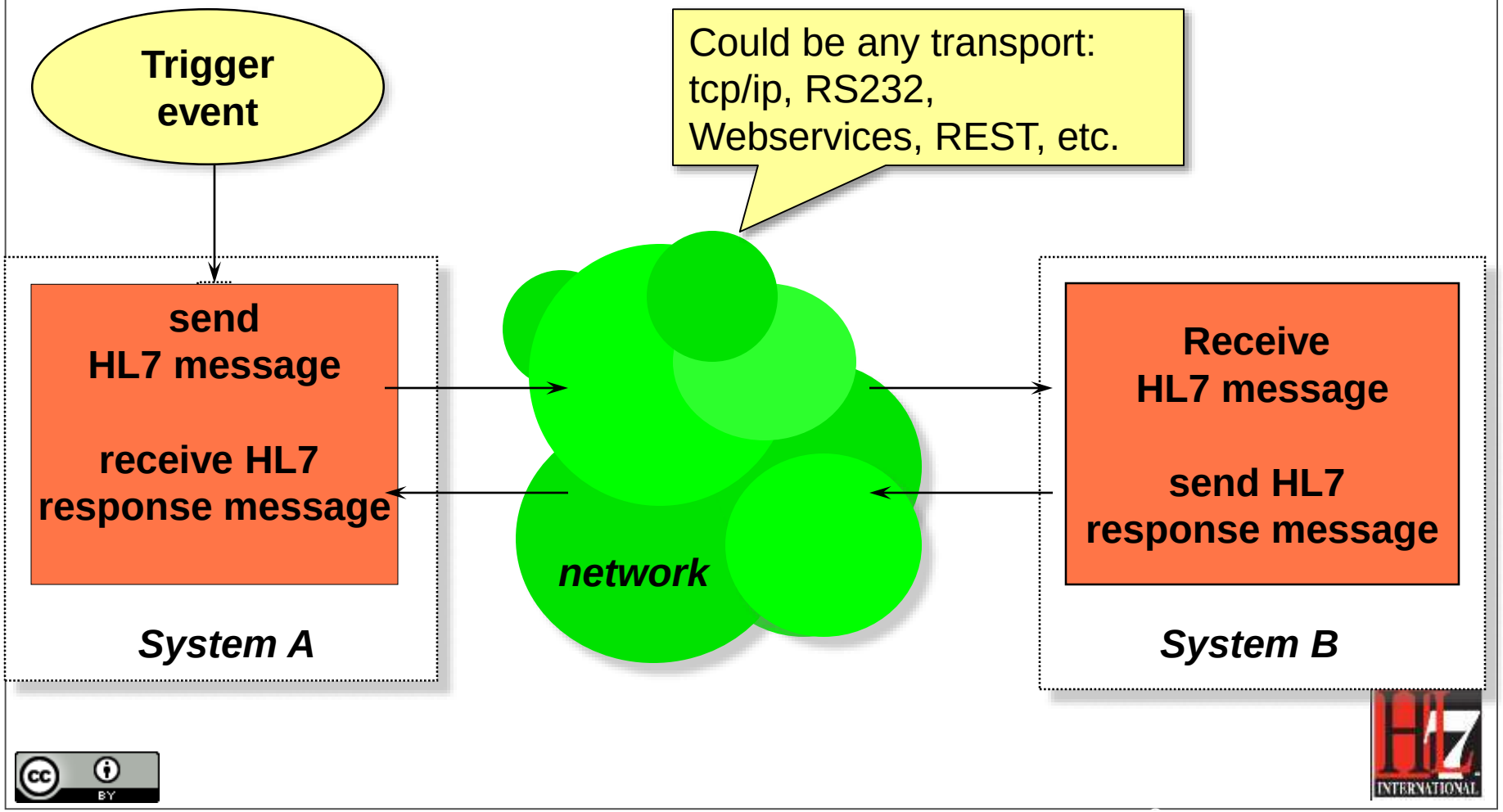
René Spronk
Tutor / Senior Consultant
Ringholm bv, the Netherlands

eMail: Rene.Spronk@Ringholm.com
Web: http://www.Ringholm.com

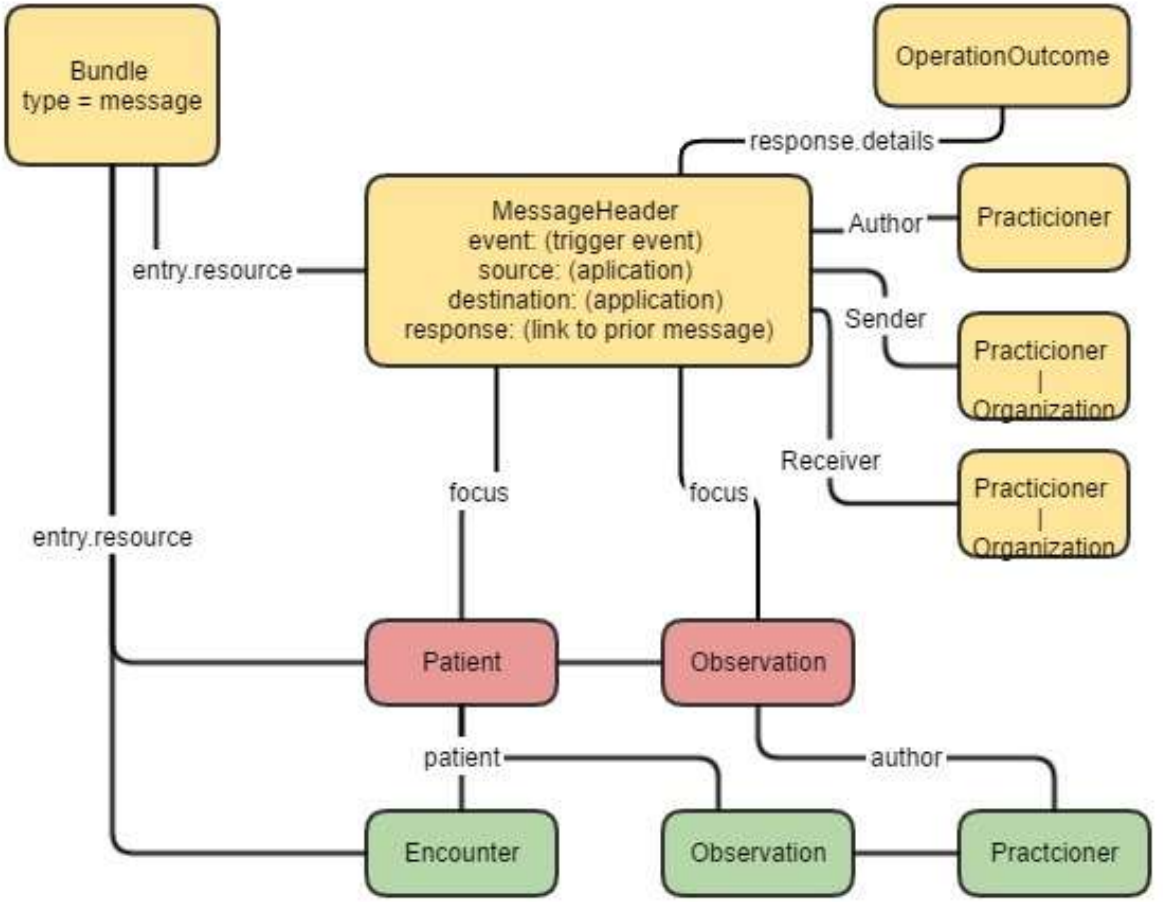
Messaging

- Loosely coupled
  - Destination may not be available
  - Source may not be queryable
  - Store and forward architecture
  - Worst case assumption – send rich set of data each and every time, receiver to select/process relevant parts
- Trigger event based
  - Trigger Event = Business reason for data exchange

Basic Principles of HL7 Messaging

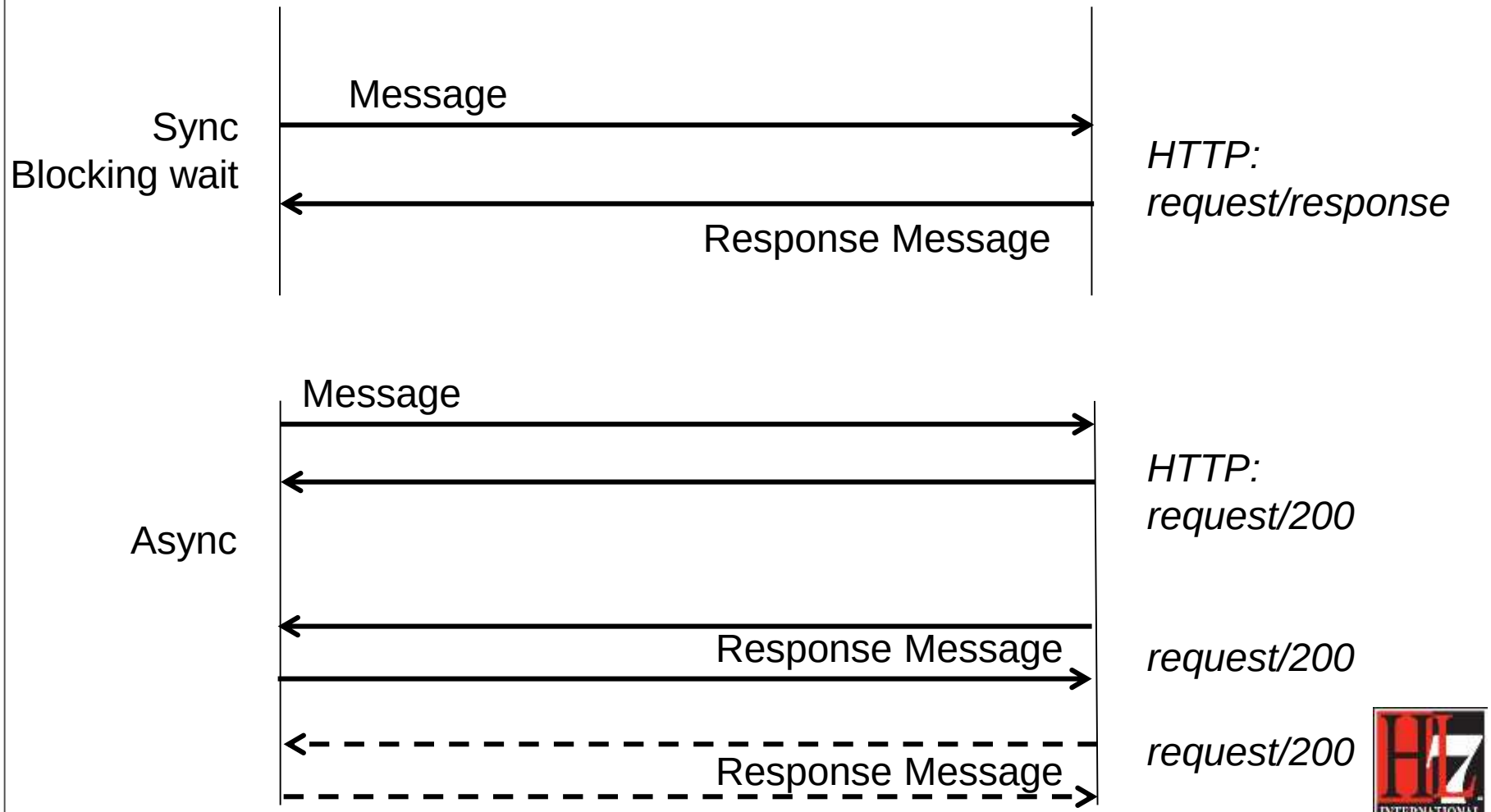


Message Structure



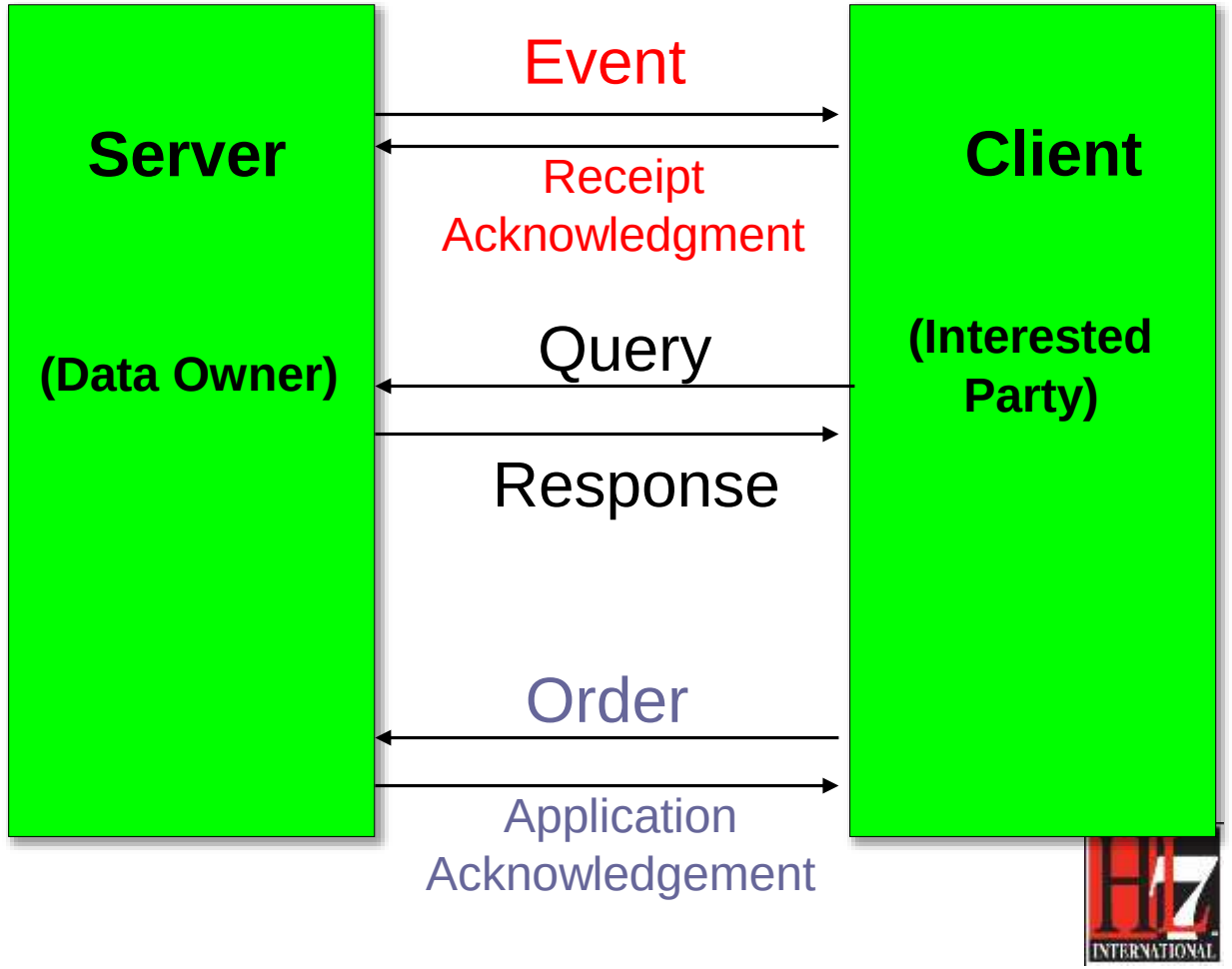
Note: v2 doesn't have the concept of focal resource. V3 does support the concept.

Sync / Async messaging patterns



Note: in v2 the use of multiple async response is exceptional.

Message Exchange Mechanisms



Note: v2 calls these Declarative, Interrogative, and Imperative

Impact of message content

Consequence	The message represents/requests a change that should not be processed more than once; e.g., making a booking for an appointment.
Currency	The message represents a response to query for current information. Retrospective processing is wrong and/or wasteful.
Notification	The content is not necessarily intended to be current, and it can be reprocessed, though there may be version issues created by processing old notifications.

Timeouts, retransmissions

- Notification/Query: resend same message after timeout. Receiver processes and creates new response.
- Message of Consequence:
  - Don’t perform same activity again, don’t process message again
  - Send ‘previous response’ from cache. Responses need to be cached for 15 minutes (at least)

Trigger Events

- Business Reason for the data exchange
- Current example values: (https://www.hl7.org/fhir/valueset-message-events.html)

Code	Category	Description	Request Resources	Response Resources
MedicationAdministration-Complete	Consequence	Change the status of a Medication Administration to show that it is complete.	MedicationAdministration	MedicationAdministration
admin-notify	Notification of a change to an administrative resource (either create or update). Note that there is no delete, though some administrative resources have status or period	Device	--	
		Device	--	
		Group	--	
		Location	--	
		--	Organization	(see Patient)
		Patient	--	
		Practitioner	--	
		Practitioner	--	
		--	RelatedPerson	

Extract, Example binding

Note: a v2 trigger event implies a certain message structure. Not so in FHIR.

Trigger Events

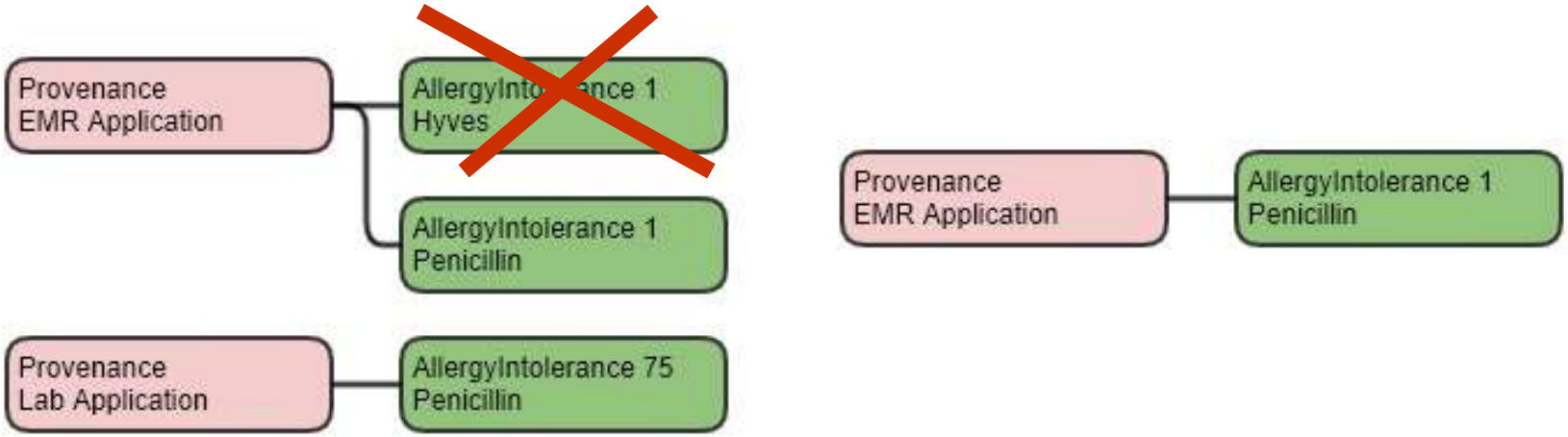
- The trigger event code list is relatively short
  - Need triggers that (in notifications) precoordinated from/to states in state changes, not just "something changed" type triggers
    - Some resources (e.g. encounter) contain an optional state history
  - Focal resource. No need to precoordinate focal resource type in the trigger event (although for conformance reasons it will probably still be done)
  - Early days, triggers to be added as implementation occurs.

Notifications: snapshot issue

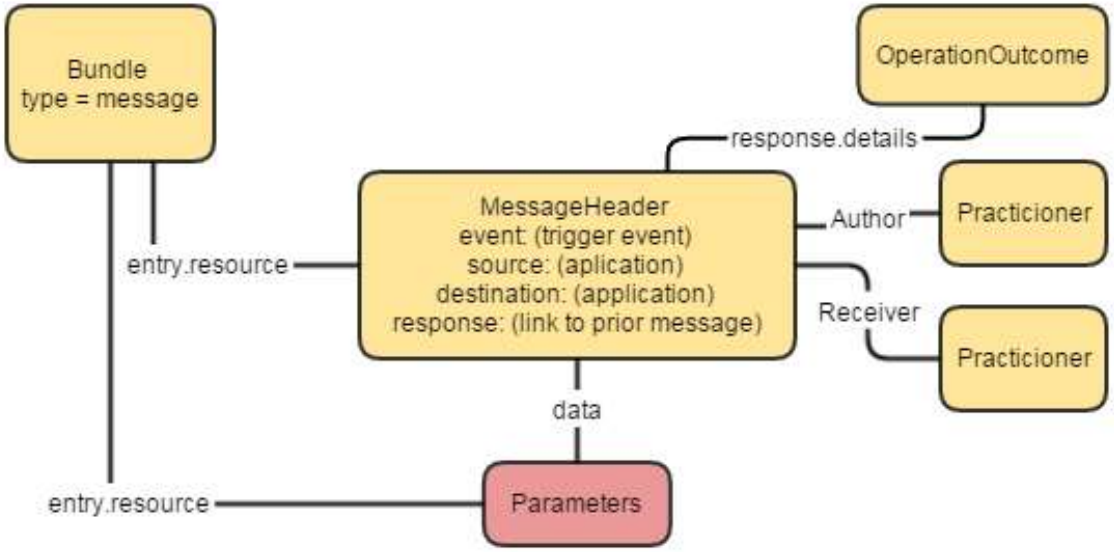
- Trigger event “here’s a list of known allergies”
  - A deleted Allergy wouldn't trigger a message saying "this is the deleted allergy" but instead a message saying "these are the active allergies, delete all previously stored data and replace with the current set". Or explicitly designate some as ‘being inactive’ (assuming sender is aware of those)
- General “snapshot” problem of messaging

Snapshot: rip & replace

- Rip and replace data *sourced from the same application*



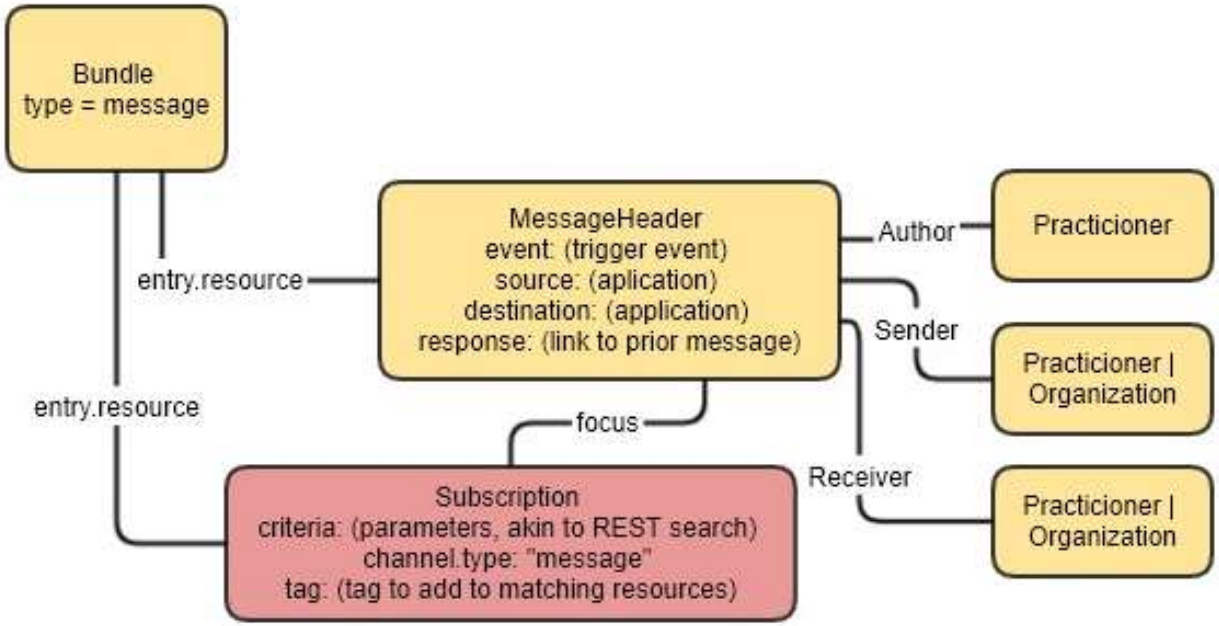
Queries



Name	Flags	Card.	Type	Description & Constraints
Parameters			Resource	Operation Request or Response
parameter	I	0..*	BackboneElement	Operation Parameter *A parameter must have a value or a resource, but not both*
name		1..1	string	Name from the definition
value[x]	I	0..1	*	If parameter is a data type
resource	I	0..1	Resource	If parameter is a whole resource
part	I	0..*	see parameter	Named part of a parameter (e.g. Tuple) *A part must have a value or a resource, but not both*

Subscription Mechanism

- Enables dynamic attachment/detachment to a messaging stream



Note: v2 does have a subscription mechanism, but it has seen little implementation

CONFORMANCE

Message is a contract

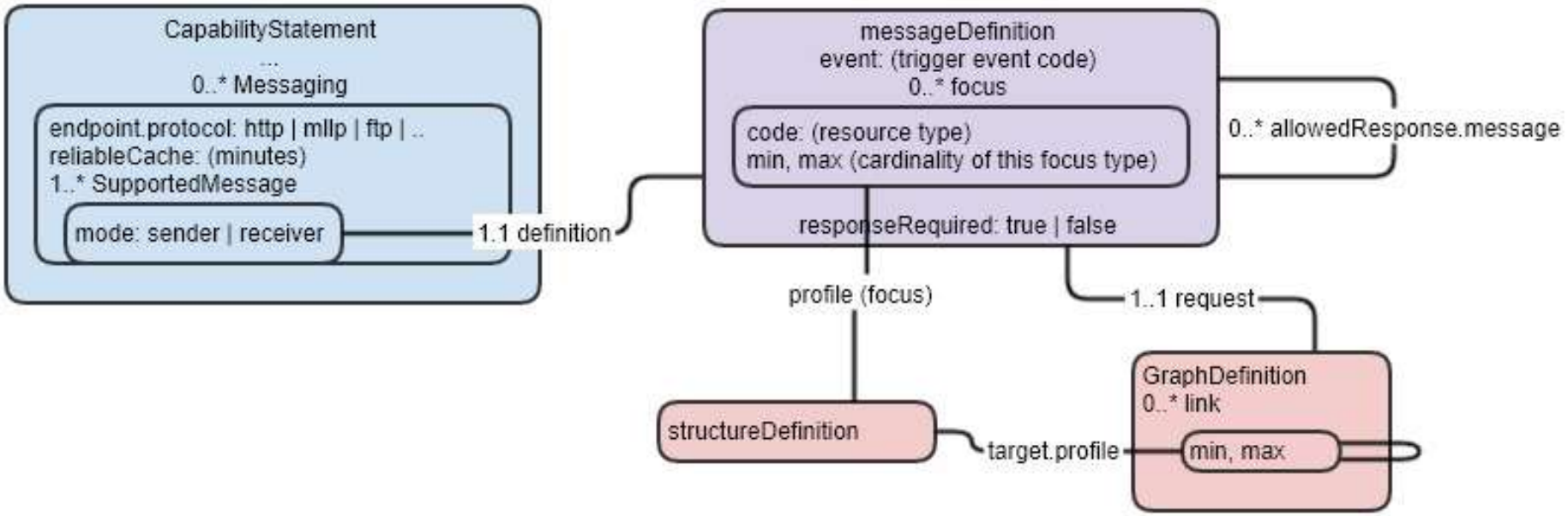
- Contract between sender and receiver
  - Content of the message (profile, implementation guide)
  - When to be sent (trigger event)
  - How to respond
  - Expected behavior of applications upon recieving a message
    - E.g. ‘Discharge notification’ : Different resulting behavior in case one is a catering application, or room cleaning, or..

MessageDefinition (excerpt)

Message = structure(graphDefinition), Trigger event (code), allowed response messages (messageDefinition)

event	Σ	0..1	uri	Link to the Event type
category	Σ	0..1	code	Consequence | Currency | Notification MessageSignificanceCategory (Required)
profile		0..1	Reference (graphDefinition)	
focus	Σ I	0..*	BackboneElement	Resource(s) that are the subject of the event *+ Max must be postive int or **
code	Σ	1..1	code	Type of resource ResourceType (Required)
profile		0..1	Reference(StructureDefinition)	Profile that must be adhered to by focus
min		0..1	unsignedInt	Minimum number of focuses of this type
max	I	0..1	string	Maximum number of focuses of this type
responseRequired		0..1	boolean	Is a response required?
allowedResponse		0..*	BackboneElement	Responses to this message
message		1..1	Reference(MessageDefinition)	Reference to allowed message definition response
situation		0..1	markdown	When should this response be used

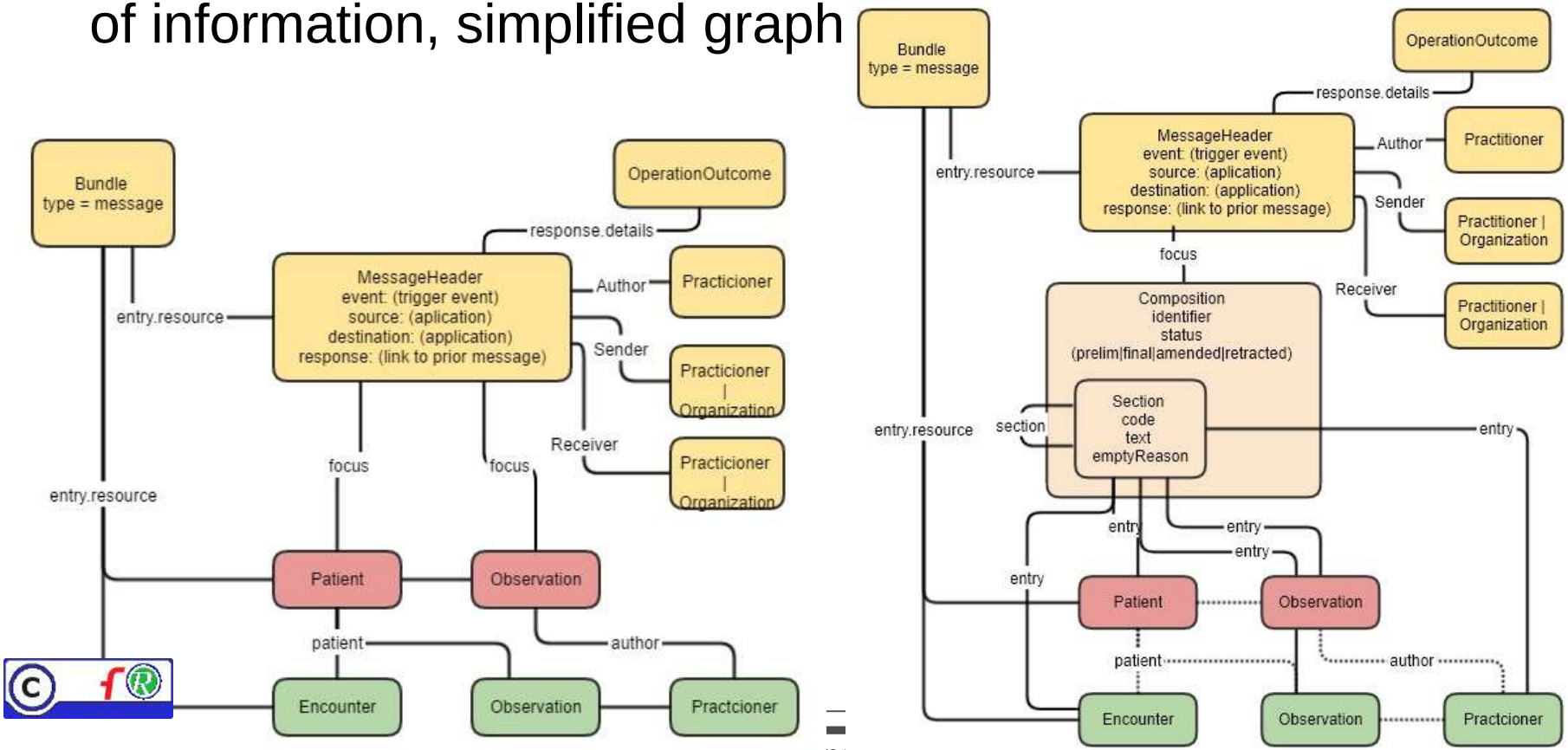
Messaging, Conformance



Note: post-STU3 model.

Composition

- Composition as the focal Resource
- Desire for message to be a complete self-referencing "unit" of information, simplified graph



PROJECTS

Canada: mailbox

- Canada is using messaging with a "mailbox" approach on a central server for secure communication and e-prescribing.
- Rationale was that recipients wouldn't necessarily be up at time of delivery and a desire to send all relevant information as a single package.
- There was no desire to support information access beyond specifically targeted recipients, so hosting the information a FHIR server for query seemed inappropriate.

Ontario, Canada

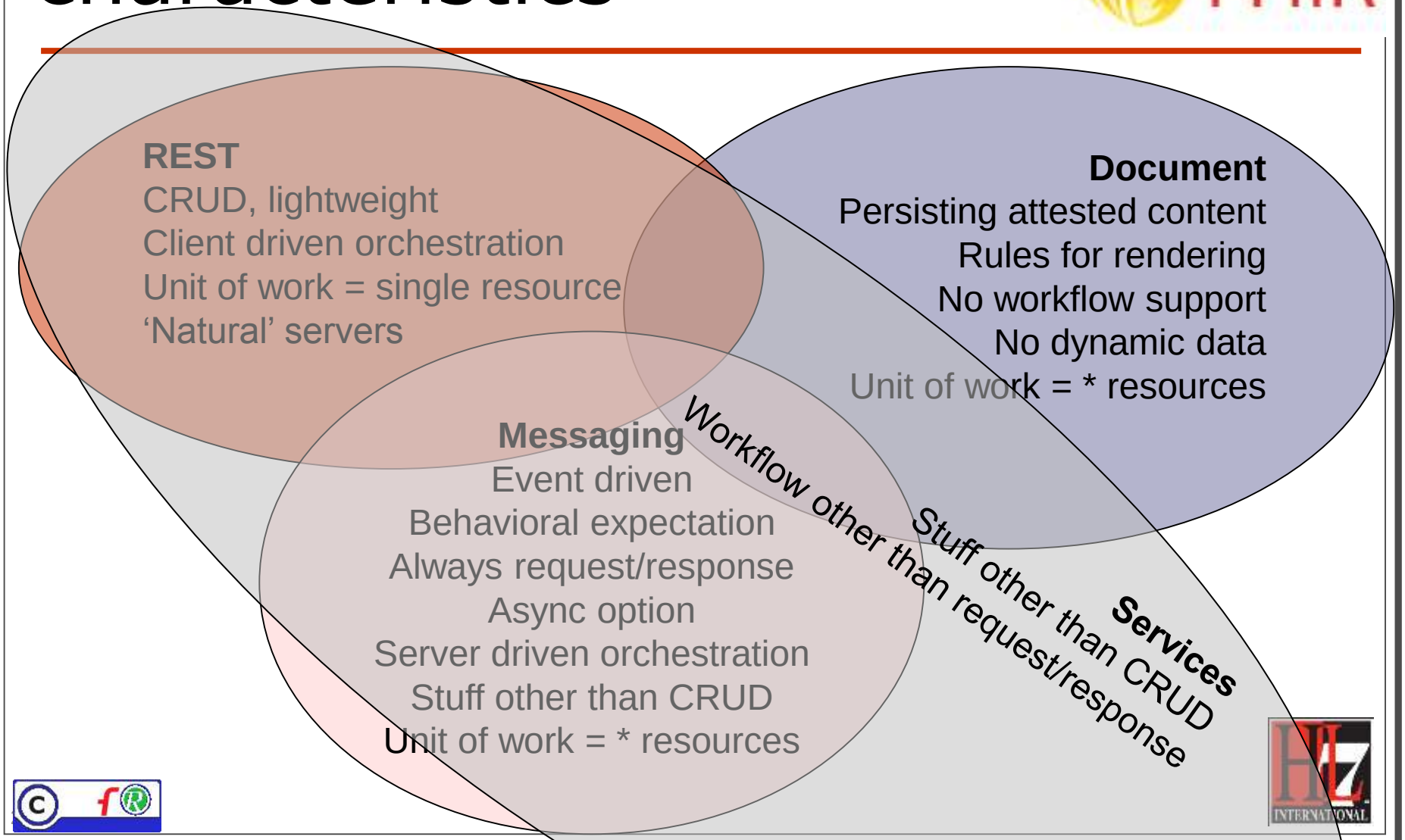
- Use messaging for submission of immunization records by doctor's portal and EMR systems to the provincial immunization repository.
- The submission goes into a temporary repository first where public health nurses review them, might call a provider in case they need to clarify anything, might add a patient record if it's not in the system yet (e.g., newborn or newcomer) and then approve these immunization records to be added to the main repository.

NHS England

- Uses messaging, REST and Documents.
- REST has volumetric concerns
  - Compare 5 million messages/day with 50 million RESTful queries
- Legacy applications:
  - not queryable
  - may not be available 24/7
  - no assumption of client orchestration,
  - no assumption that there is a FHIR aware backend/storage

SUMMARY

Paradigms - characteristics

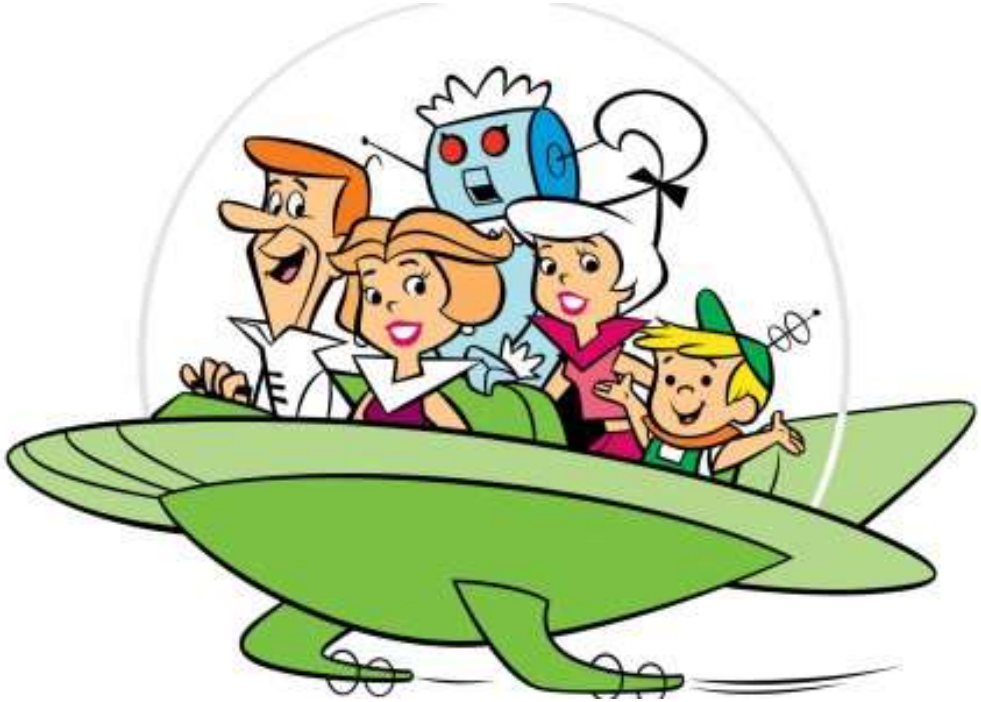


Debunking Messaging Myths

- Messaging is old fashioned
- Nobody uses FHIR messaging
- Messaging is just an async way of transferring data between REST endpoints
- REST can do this as well, and more

Messaging, summary

- Loosely coupled
  - Destination may not be available
  - Source may not be queryable
  - Store and forward architecture
  - Worst case assumption – send rich set of data each and every time, receiver to select/process relevant parts; “snapshot”
    - Server driven orchestration
- Event driven
- Behavioral expectation
- Always request/response
- Async option



QUESTIONS?