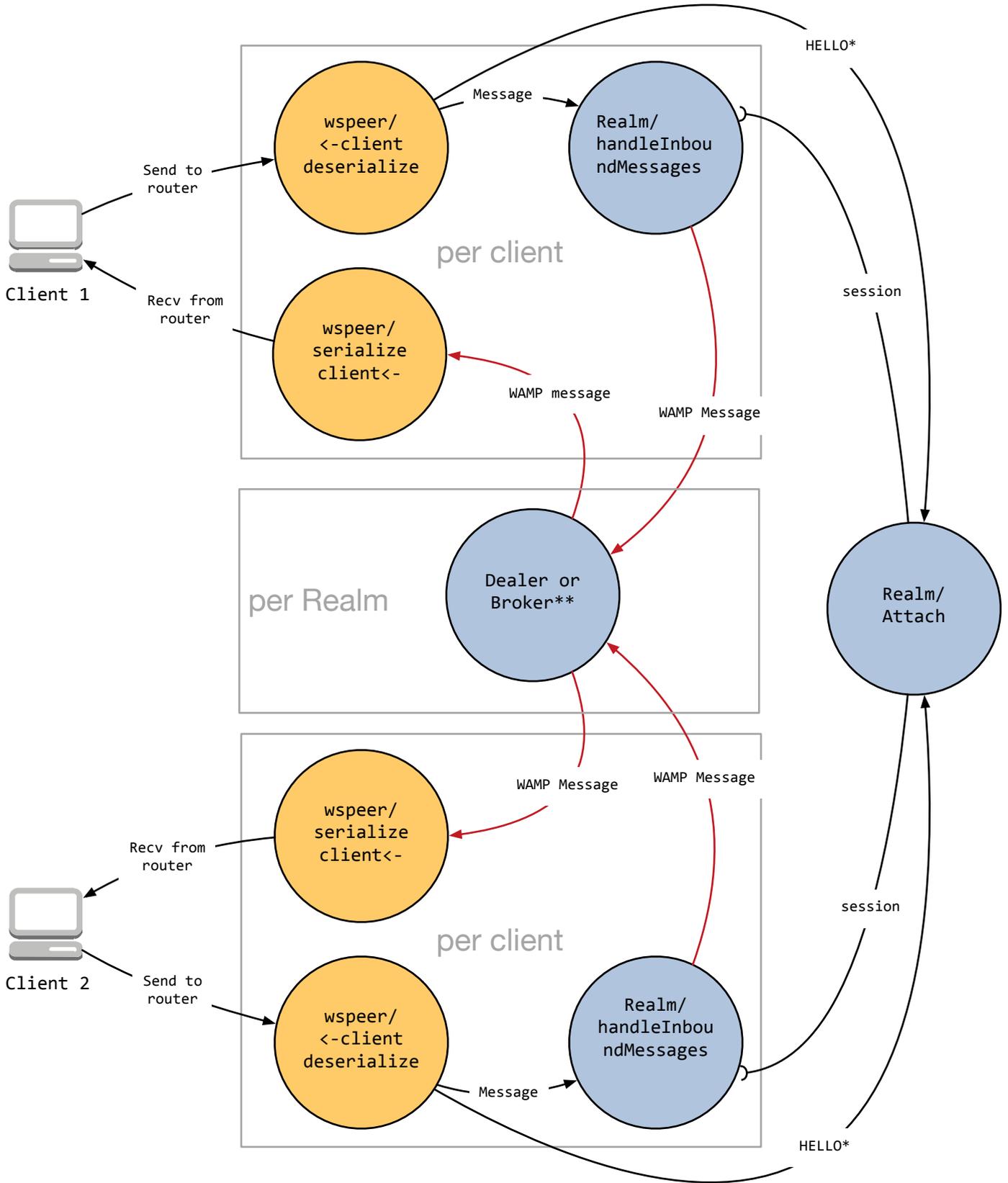


Nexus WAMP Router

Coroutine/Channel Floorplan
External Clients Only

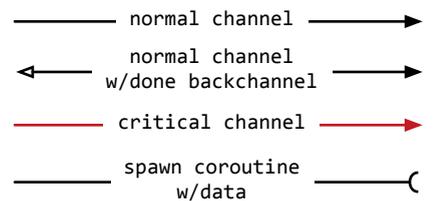


* HELLO is only message sent to Realm. WELCOME response not shown.

** Dealer and Broker use separate coroutines (not shown separately)

Created by server transport

Created by Router/Realm



Nexus WAMP Router

Coroutine/Channel Floorplan
External Clients Only

Realm/ Attach:

- * Lifetime: long-running (generally lifetime of router)
- * Inputs: HELLO message for new clients
- * Outputs: WELCOME or ERROR to new clients
- * Responsibilities:
 - Records sessions attached to realm
 - Spawns client message handler for each accepted client
 - Spawns single meta API client to publish meta events and handle meta procedures

Dealer/Broker:

- * Lifetime: long-running (lifetime of Realm)
- * Inputs: Messages from source clients
- * Outputs: Messages to destination clients
- * Responsibilities:
 - Broker: Subscribe clients, publish events
 - Dealer: Register callee, call RPC, cancel RPC
- * Special considerations:
 - Routing done on critical channel/coroutine (most avoid non-critical work)
 - Downstream channel must never block
 - Concurrent with and independent of Brokers and Dealers in other Realms

Realm/ handleInboundMessages:

- * Lifetime: As long as client connected to router
- * Inputs: WAMP Messages from clients to router
- * Outputs: Validated messages to Broker or Dealer, or GOODBYE or ERROR to client
- * Responsibilities:
 - Reads messages from client inbound channel.
 - Validates message content.
 - Calls authorizer to authorize message for routing.
 - Forwards message to Dealer or Broker for routing.

wspeer/ <-client:

- * Lifetime: As long as client connected to router
- * Inputs: Binary messages from websocket (or other transport)
- * Outputs: WAMP messages
- * Responsibilities:
 - Read message from transport
 - Deserialize message and convert to structured WAMP message.
- * Special considerations:
 - Operates concurrently with transport writer

wspeer/ client->:

- * Lifetime: As long as client connected to router
- * Inputs: WAMP messages from router
- * Outputs: Binary messages to websocket (or other transport)
- * Responsibilities:
 - Serialize WAMP message to binary message
 - Write message to transport
- * Special considerations:
 - Operates concurrently with transport reader
 - Must never block: queue messages and drop if queue full
 - Queue size should big enough to handle message burst.