

Working Towards Interoperable Secure Messaging

This page is a collection of ideas that represent a work in progress and a place to document ideas and diagrams about the process of developing a common secure messaging environment in the Australian HL7v2 environment. It is not a specification but a place to present ideas which may help in the development of a common specification. At this time error handling is not represented. The mechanism that a store and forward system uses to deliver the message is out of scope and the server and client portions of these systems are merged into one. The methods are asynchronous on store and forward systems however.

Notes on ACKs

One issue is that when you send an ACK to a realtime provider there is not real response to this and if you are not careful you can easily develop and endless loop of ACKs. Sending of acks happens when the original ack fails to be returned in the real time connection (ie the connection is lost after the message has been received) It also happens when application ACKs are sent back at a later stage. When an ACK is sent to a realtime system we just respond with "OK" which is sent back encrypted and signed. This indicates the ack was received but avoids the endless loop issue. With store and forward messaging this is probably best done with a specific method.

Colour coding

HL7 messages are blue and non HL7 messages are red. Some are plain text and some are encrypted.

Diagram 1. To draw the diagrams these were the methods I needed to add to the actors

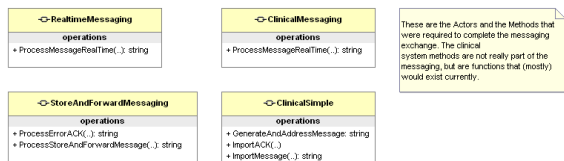


Diagram 2. This example is pure real time messaging

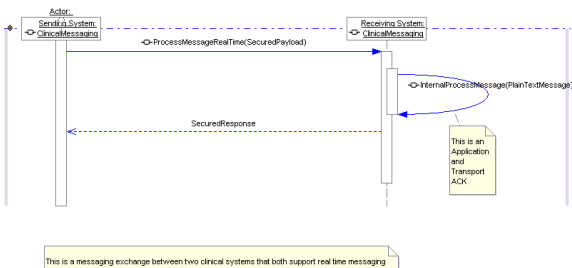


Diagram 3. This is a Clinical system sending to a realtime messaging provider

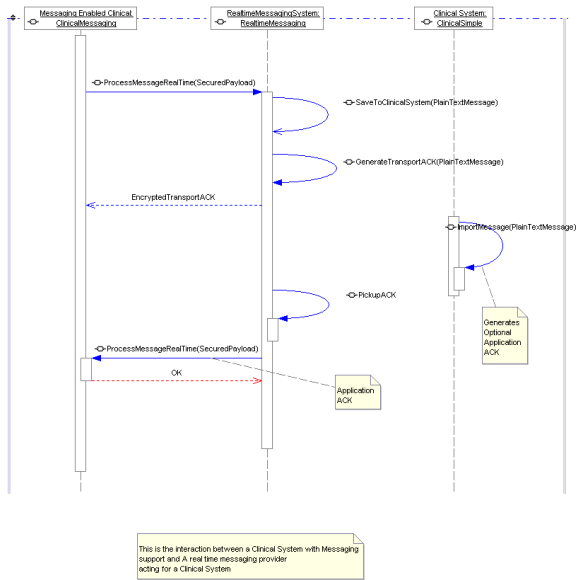


Diagram 4. This is a Clinical system sending to a Store and forward Provider

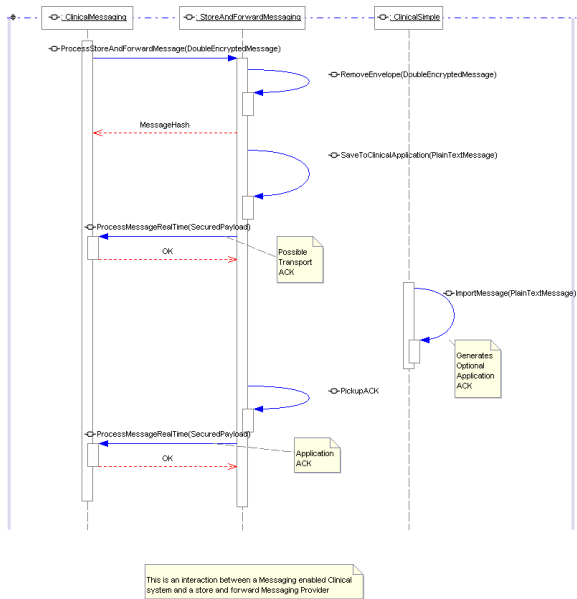


Diagram 5. This is an example interaction between 2 messaging providers

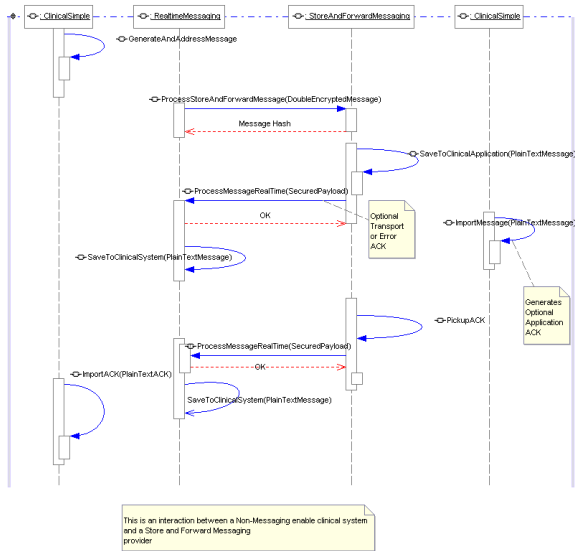


Diagram 6. This is another example of an interaction between two messaging Providers

