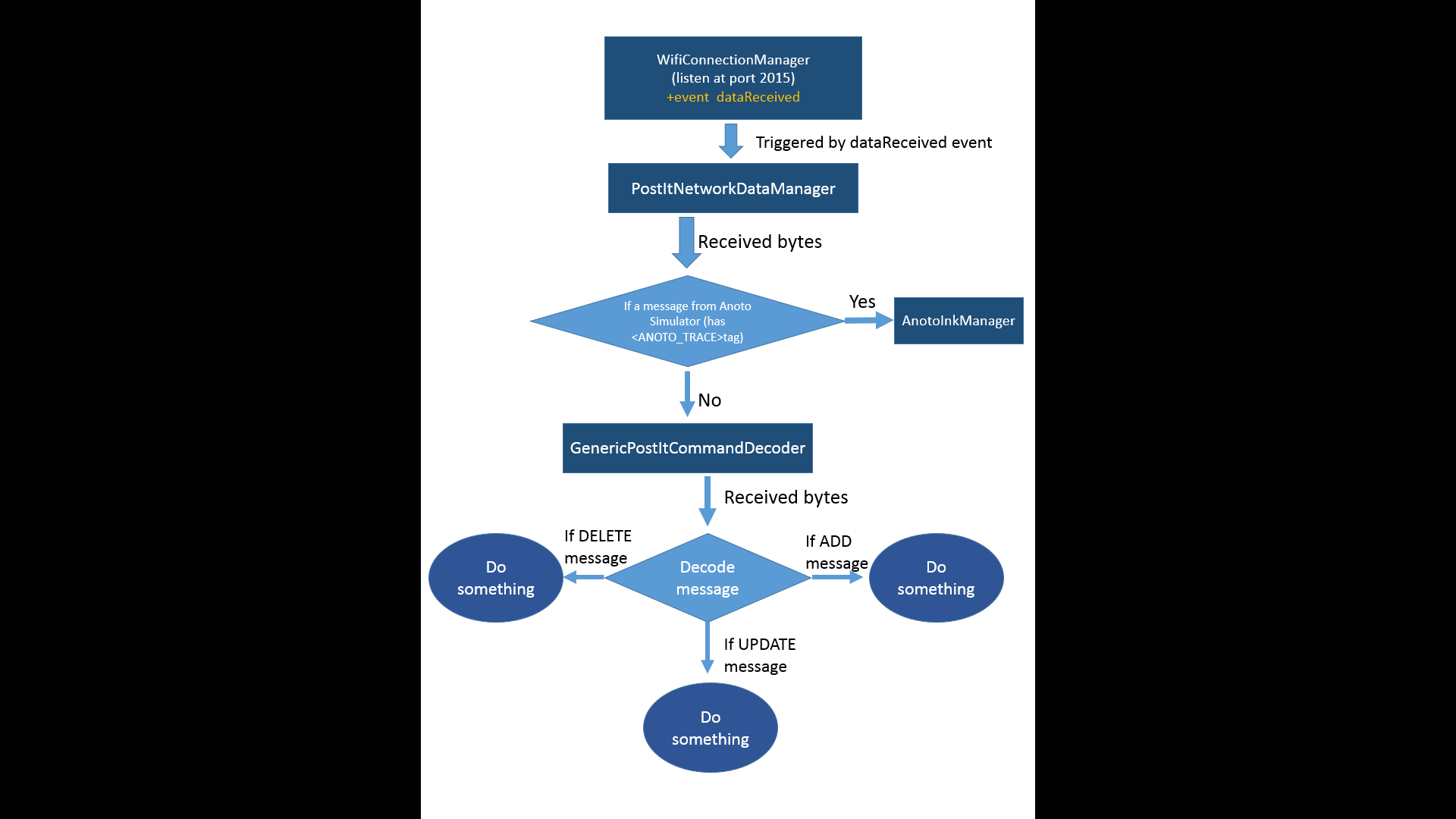
The following diagram shows the general mechanism of how the PostIt\_Prototype1 app handles data coming from other devices through Wifi connection.



In the BrainstormCanvas class (Presentation/BrainstormCanvas.cs file):

* Object *wifiManager* is an instance of **WIFIConnectionManager**. This object is listener for client requests and messages on the *port 2015*. Clients need to actively connect to the application at its IP address and port 2015. At this stage, due to security reasons, the system does not work with eduroam network.
* Once wifiManager receives data from clients, the event handler *dataReceivedHandler* will be triggered. This event is delegated to the method *networkReceivedDataHandler* of *networkDataManager*, which is an instance of **PostItNetworkDataManager**.
* In **PostItNetworkDataManager**, currently there are two objects that will handle the data received from the network depending on the type of the data.
  + *\_anotoInkManager* which is an instance of *AnotoInkManager*: this object is specially to handle data coming from the simulation application of Anoto on mobile device. If the message from client contains the tag <ANOTO\_TRACE>, it will be recognized as Anoto simulator’s message and processed by \_anotoInkManager. Otherwise, it will be handled by *\_genericPostItDecoder*.
  + *\_genericPostItDecoder* is an instance of **GenericPostItCommandDecoder**. At this moment, any messages which is not sent by the mobile Anoto simulator will be handled by this object. Such messages should belong to one of the following 3 types of message
    - ADD message: this message is enclosed by the pair of tags <ADD> and </ADD>. This message is to notify the system about the addition of a new Post It note. The message contains the ID, position and content of the PostIt note. If this message is sent by the Tabletop, the content is the image of the physical note captured by the tabletop. The format of message’s content is:

<ADD> ID X Y Orientation Size DataType Data </ADD>

* + - UPDATE message: this message is enclosed by the pair of tag <UPD> and </UPD>. Now this message is only used to tell the system if there is any change of location of existing notes. This message contains ID and new position of the updated notes.
    - DELETE message: this message is enclosed by the pair of tag <DEL> and </DEL>. This message contains the ID of the note to be removed from the server (the brainstorming whiteboard).

At this moment, the \_genericPostItDecoder is used to handle message from the tabletop. If we want to plug the Livescribe pen + Evernote in the system, a simple solution is to make Evernote send an ADD message where DataType is pdf for example and Data is the pdf file’s data. Then I’ll write a code to process the pdf file to extract the content.