Controll Developement

Erik Jonsson Thorén

March 18, 2013

Introduction

This document specifies the goals for the Controll project. It also includes some diagrams over different models and message routing specifications.

`http://github.com/thecopy/Controll`

Milestones

Note that none of these versions have been released as of this date and this document is just a weak specification.

Controll 0.0.1

The 0.0.1 version of Controll will be a prototype of the fundamental communications between a client and a zombie.

- Simple Authentication.
- Activate simple activity[1] from a client
- Register Zombie
- Register User
- Fetch Zombie Information
  - Owned zombies
  - Installed Activities on specified zombie
- Ping zombie (through server)

Controll 0.0.2

Changes from 0.0.1:

- Message queue, MQ, offline support[2]
- Message delivery verification[3]

See appenix A2 for a specification of the message routing in 0.0.2 for a clear overview of how the offline queue support is specified to be implemented.

[1] A terminating response from the activity. No further dialog between client and the activity.

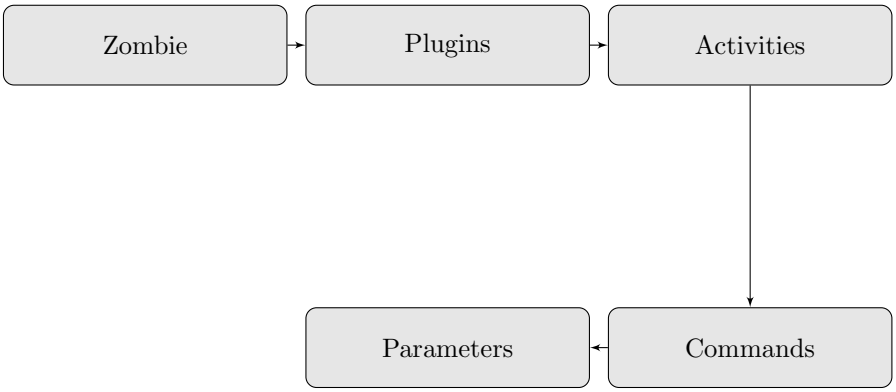
[2] A message to a zombie or a client will wait until the zombie or client becomes online and then deliver the message

[3] Ties in very neatly with the MQ offline support

Appendix

A1. The Zombie Model

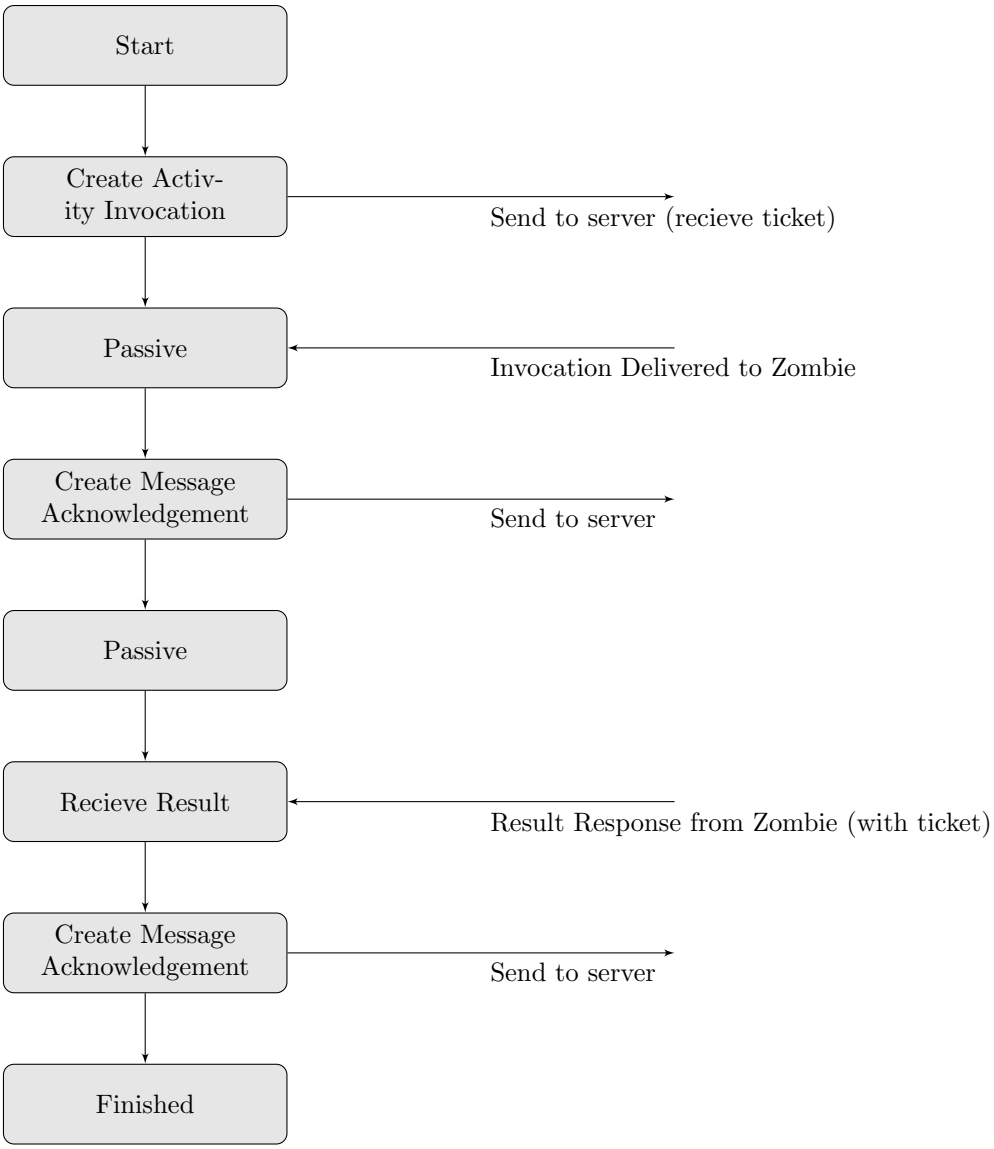
The arrow represents *have many.*



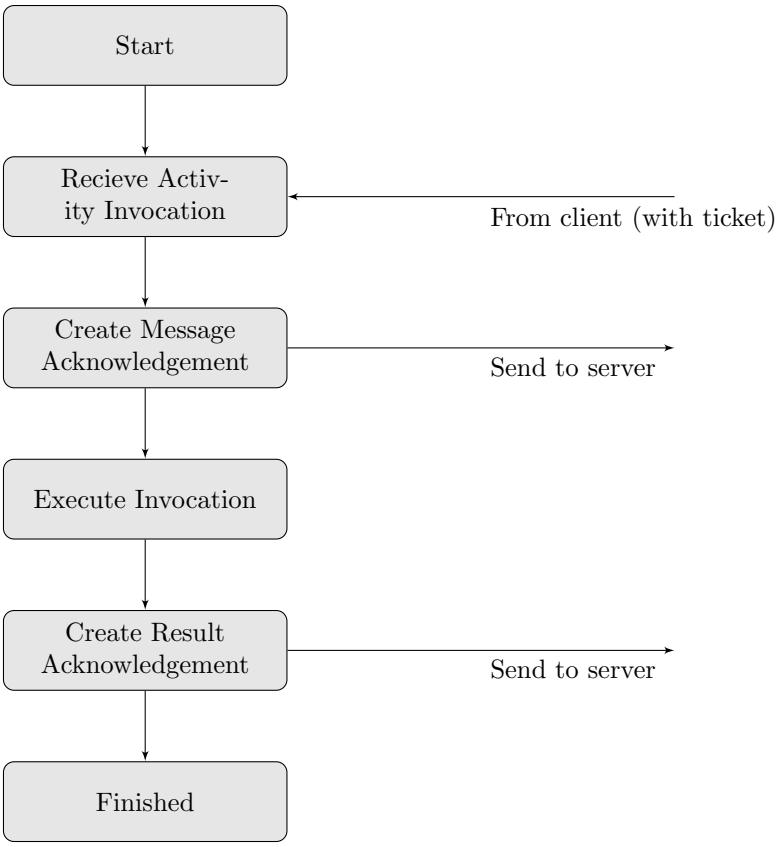
A2. The messaging specification model in 0.0.2

These three flow charts shows the specification for the message routing between the client, the server and the zombie when the user invokes an activity on the zombie.
Note: Due to the implementation of SignalR (the abstracted transport) we will always get an acknowledgement whether what we send to the server is received or not.

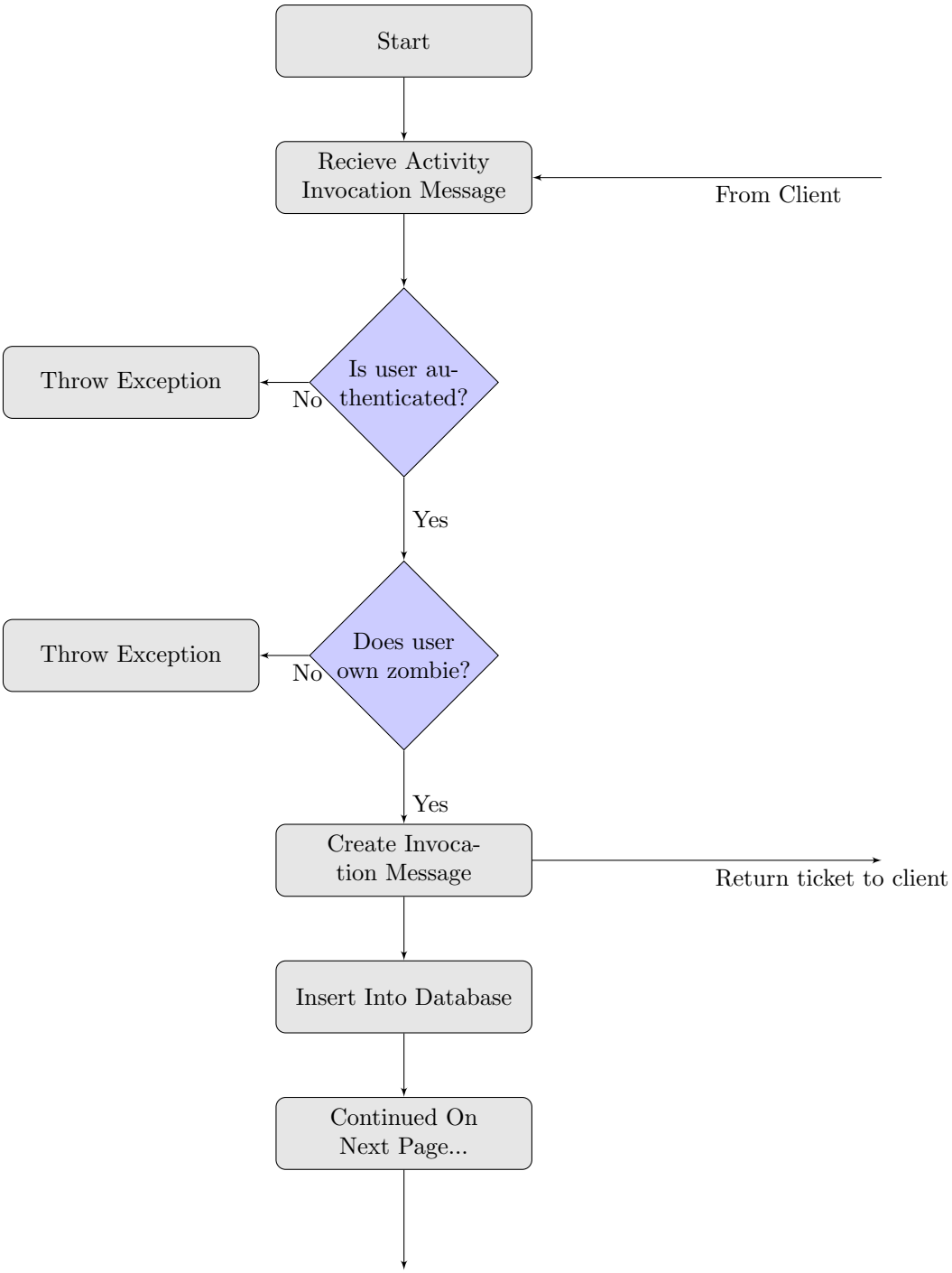
The Client

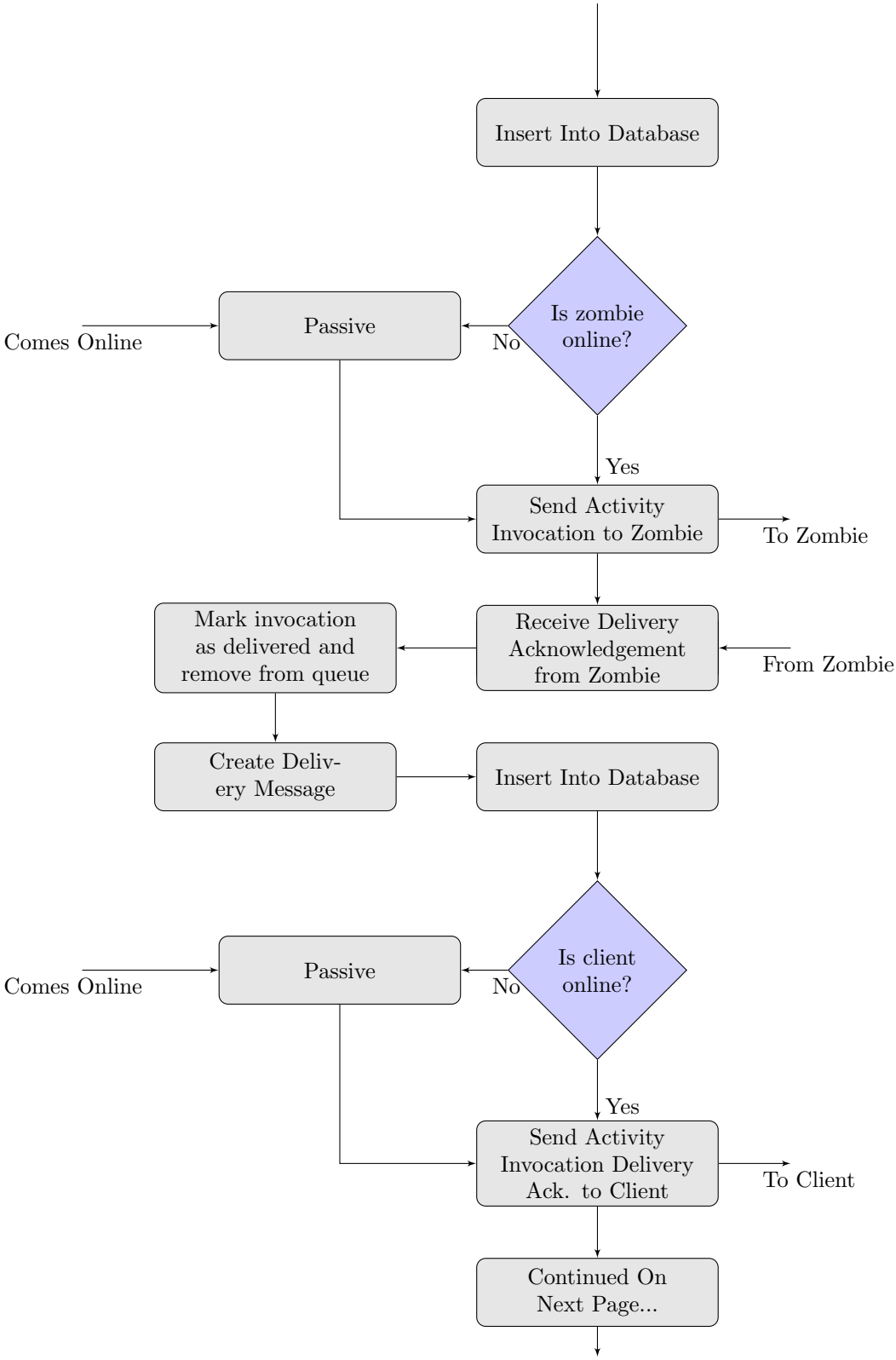

Start
Create Activity Invocation
Send to server (recieve ticket)
Passive
Invocation Delivered to Zombie
Create Message Acknowledgement
Send to server
Passive
Recieve Result
Result Response from Zombie (with ticket)
Create Message Acknowledgement
Send to server
Finished

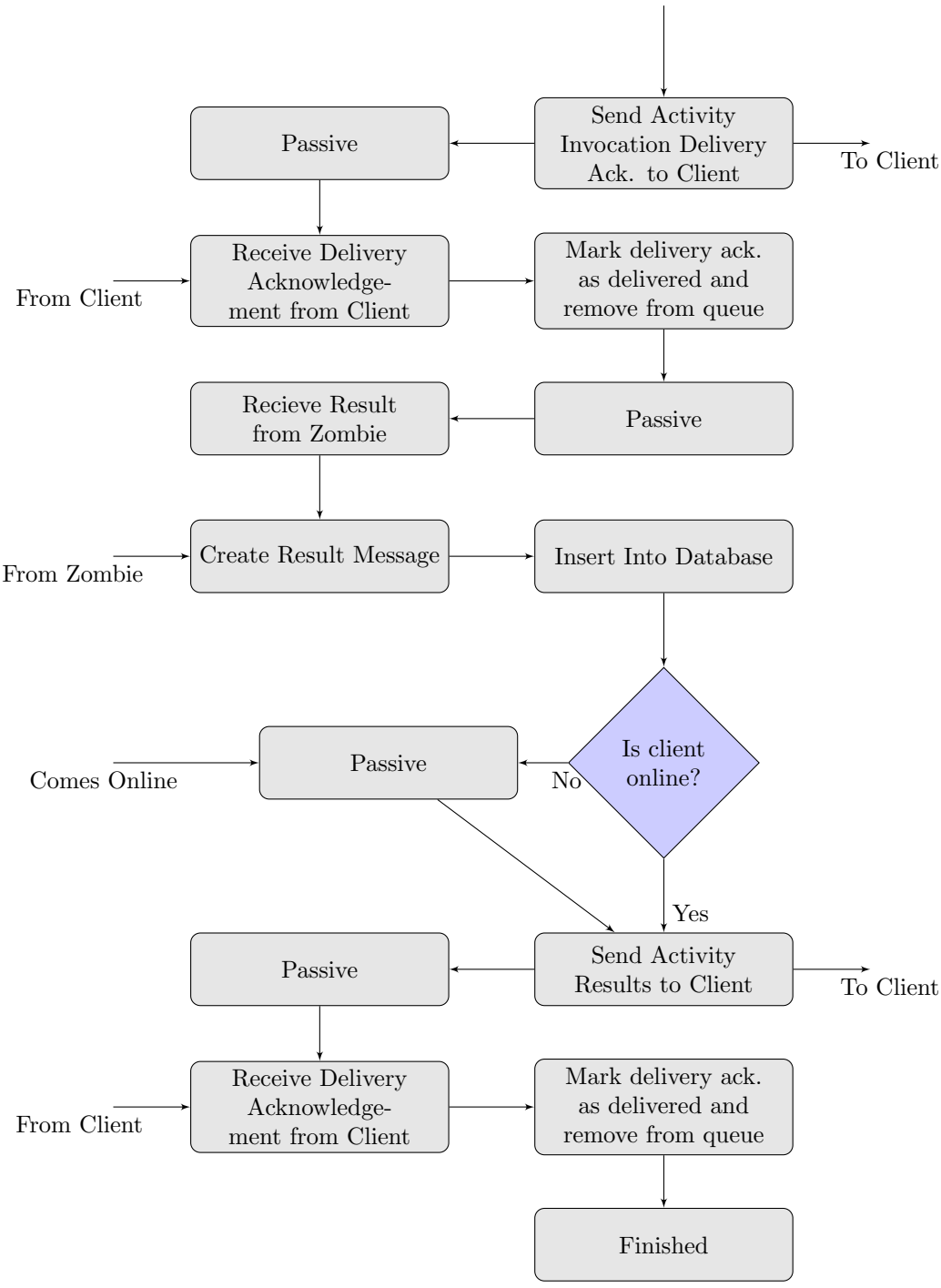
The Zombie


Start
Recieve Activity Invocation
From client (with ticket)
Create Message Acknowledgement
Send to server
Execute Invocation
Create Result Acknowledgement
Send to server
Finished

The Server


Start
Recieve Activity Invocation Message
From Client
Is user authenticated?
No
Throw Exception
Yes
Does user own zombie?
No
Throw Exception
Yes
Create Invocation Message
Return ticket to client
Insert Into Database
Continued On Next Page...


Insert Into Database
Is zombie online?
No
Passive
Comes Online
Yes
Send Activity Invocation to Zombie
To Zombie
Receive Delivery Acknowledgement from Zombie
From Zombie
Mark invocation as delivered and remove from queue
Create Deliv- ery Message
Insert Into Database
Is client online?
No
Passive
Comes Online
Yes
Send Activity Invocation Delivery Ack. to Client
To Client
Continued On Next Page...


Send Activity Invocation Delivery Ack. to Client
To Client
Passive
Receive Delivery Acknowledge- ment from Client
From Client
Mark delivery ack. as delivered and remove from queue
Passive
Recieve Result from Zombie
Create Result Message
From Zombie
Insert Into Database
Is client online?
No
Passive
Comes Online
Yes
Send Activity Results to Client
To Client
Passive
Receive Delivery Acknowledge- ment from Client
From Client
Mark delivery ack. as delivered and remove from queue
Finished