Tom's Blog -

ZeroMQ here we go

MARKDOWN At the moment I'm working on a more distributed project. One central thing is that I want a **local-server** to be control from the **main-server** that is located somewhere else. (e.g. the internet) I'm using thrift-services and want on the one side to be able that the **local-server** can call the service-layer of the main-server and on the other side I want the **main-server** to be able to call the service-layer on the **local-server**. *'Real'* clients connect via a separate port to the server's Client-Service-Layer. I actually have a solution. I share the socket that is created by the local server when it connects to the main-server. Meaning for the main-server: * incoming-data goes to the **main-server**-service-implementation * outgoing-data is mapped to a client of the **local-server-thrift-client** The problem with this approach, you are forced to use oneway-calls in your thrift-service, which is a bit hacky if you want to give back information or want to throw exceptions. Otherwise you would have to label the traffic so everything is sent to the right requesting calls. (EDIT: *Another approach could have been to open a second connection from client to server and to use on for the incoming and the other for the outgoing traffic*)
![current](images/vdj-thrift.png) Well, the oneway-call-approach works, nonetheless I have had the feeling that it is a bit too hacky. And since [ZeroMQ](http://zeromq.org/) is on my radar for some time already I throw the thrift-transport to the trash and create one based on ZeroMQ. Good chance to use it in a real project instead of just doing some samples or tutorials. Btw, the [ZeroMQ-guide](http://zguide.zeromq.org/page:all) is very good to read and quite good to follow and even funny ;) Here some quotes: * *...TCP has a long timeout (30 minutes or so), that means that it can be impossible to know whether a peer has died, been disconnected, or gone on a weekend to Prague with a case of vodka, a redhead, and a large expense account....* * *...Let's see an example that pushes out weather updates consisting of a zip code, temperature, and relative humidity. We'll generate random values, just like the real weather stations do....* * *...What it means is that it's safe to repeat an operation. Checking the clock is idempotent. Lending ones credit card to ones children is **not**...* Links: [Some read about thrift bidirectional-connection](http://joelpm.com/2009/04/03/thrift-bidirectional-async-rpc.html) [ZeroMQ-Guide](http://zguide.zeromq.org/page:all) PS: Btw, this is the first blog-post I write with my markdown-plugin for pebble. (This should be a post of its own)

Published by Thomas Trocha
Wed Dec 28 04:02:00 CET 2016
http://thomas.trocha.com:80/pebble/zeromq-here-we-go