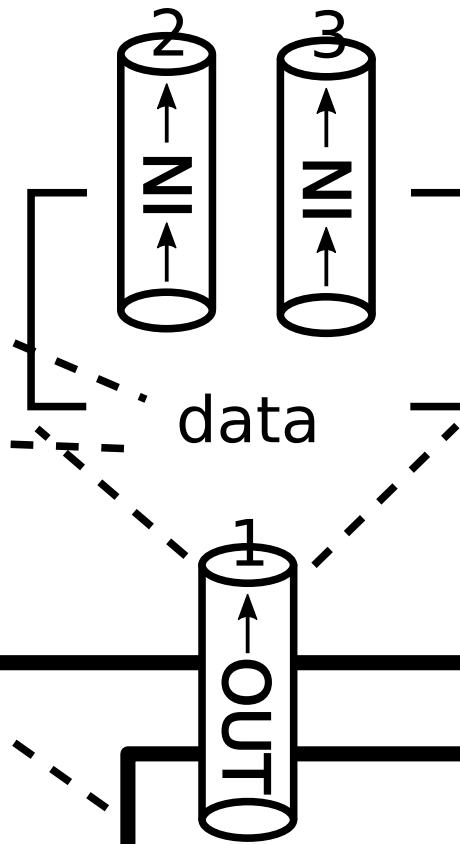


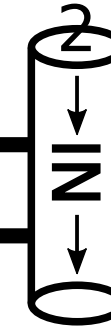
thread: distribute_data(devices)

Loop over all devices to distribute data (~100Hz)

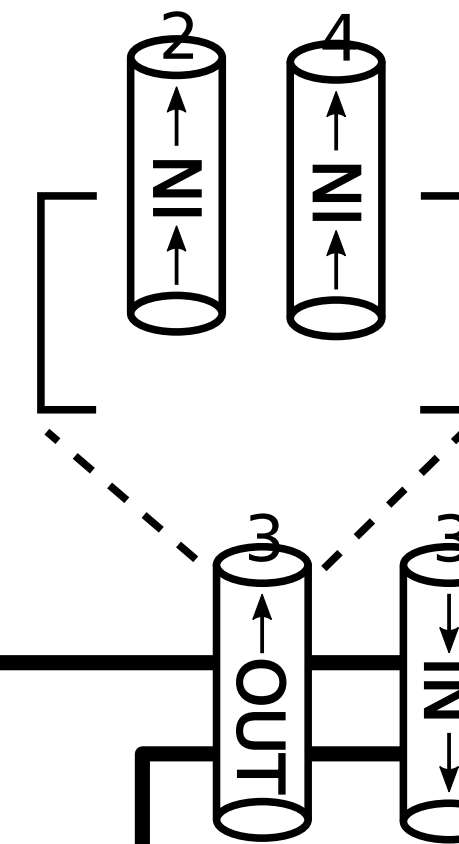
```
data = {} # Dictionary
data['t'] = time.time() # Unix time
# redvypr.distribute_data adds (if not existing)
data['host']
data['device']
```



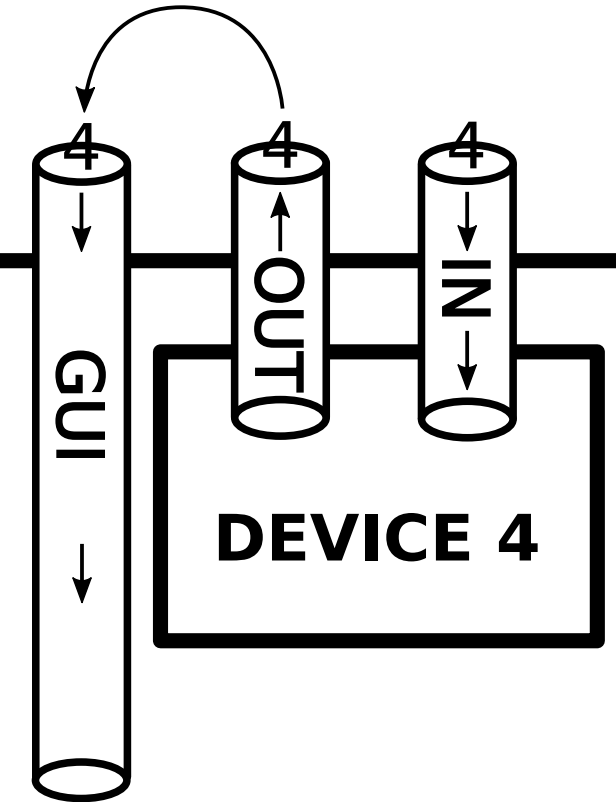
DEVICE 1



DEVICE 2



DEVICE 3



DEVICE 4

```
def distribute_data(devices):
    """ The heart of redvypr, this functions distributes the queue data onto the subqueues.
    """
    while True:
        time.sleep(0.01)
        for devicedict in devices:
            device = devicedict['device']
            while True:
                try:
                    data = device.dataqueue.get(block=False)
                    devicedict['numdata'] += 1
                except Exception as e:
                    break

            # Add deviceinformation to the data package
            if('device' not in data.keys()):
                data['device'] = str(device.name)
                data['host'] = hostinfo

            if True:
                for dataout in devicedict['dataout']:
                    devicedict['numdataout'] += 1
                    try:
                        dataout.put_nowait(data)
                    except Exception as e:
                        logger.debug(funcname + ':dataout of :' + devicedict['device'].name + ' full')
                for guiqueue in devicedict['guiqueue']: # Put data into the guiqueue, this queue does always exist
                    try:
                        guiqueue.put_nowait(data)
                    except Exception as e:
                        logger.debug(funcname + ':guiqueue of :' + devicedict['device'].name + ' full')
```

```
def start(datainqueue,dataqueue,comqueue):
    while True:
        try:
            com = comqueue.get(block=False)
            break
        except:
            pass

        time.sleep(0.05)
        while(datainqueue.empty() == False):
            try:
                data = datainqueue.get(block=False)
                dataqueue.put(data)

            except Exception as e:
                pass
```

thread: DEVICE 3