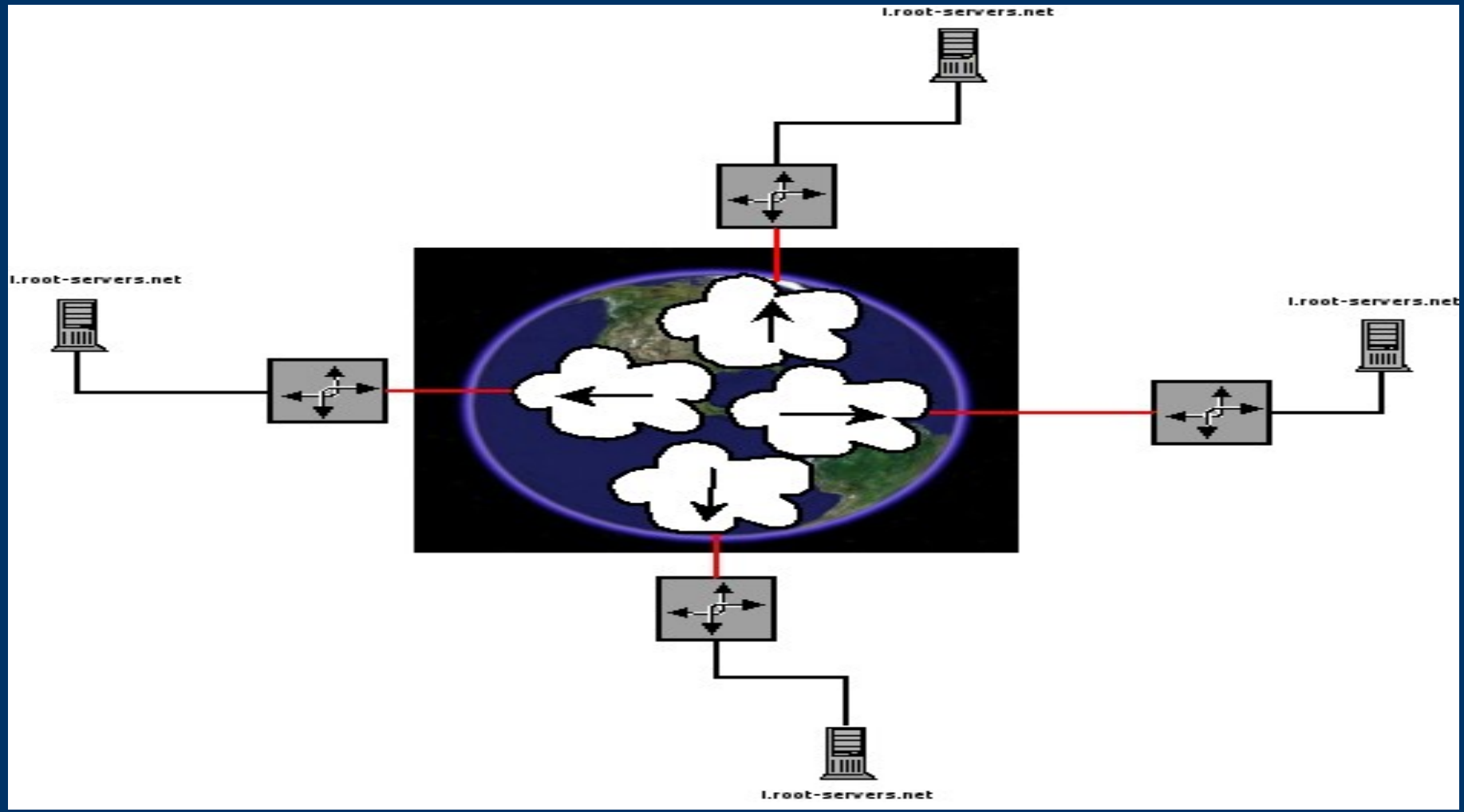
DSC-Mithril

Autonomica AB
Carl Olsen
(calle@autonomica.se)

l.root-servers.net anycast cluster

- Multiple locations
- Decentralized service

l.root-servers.net anycast cluster



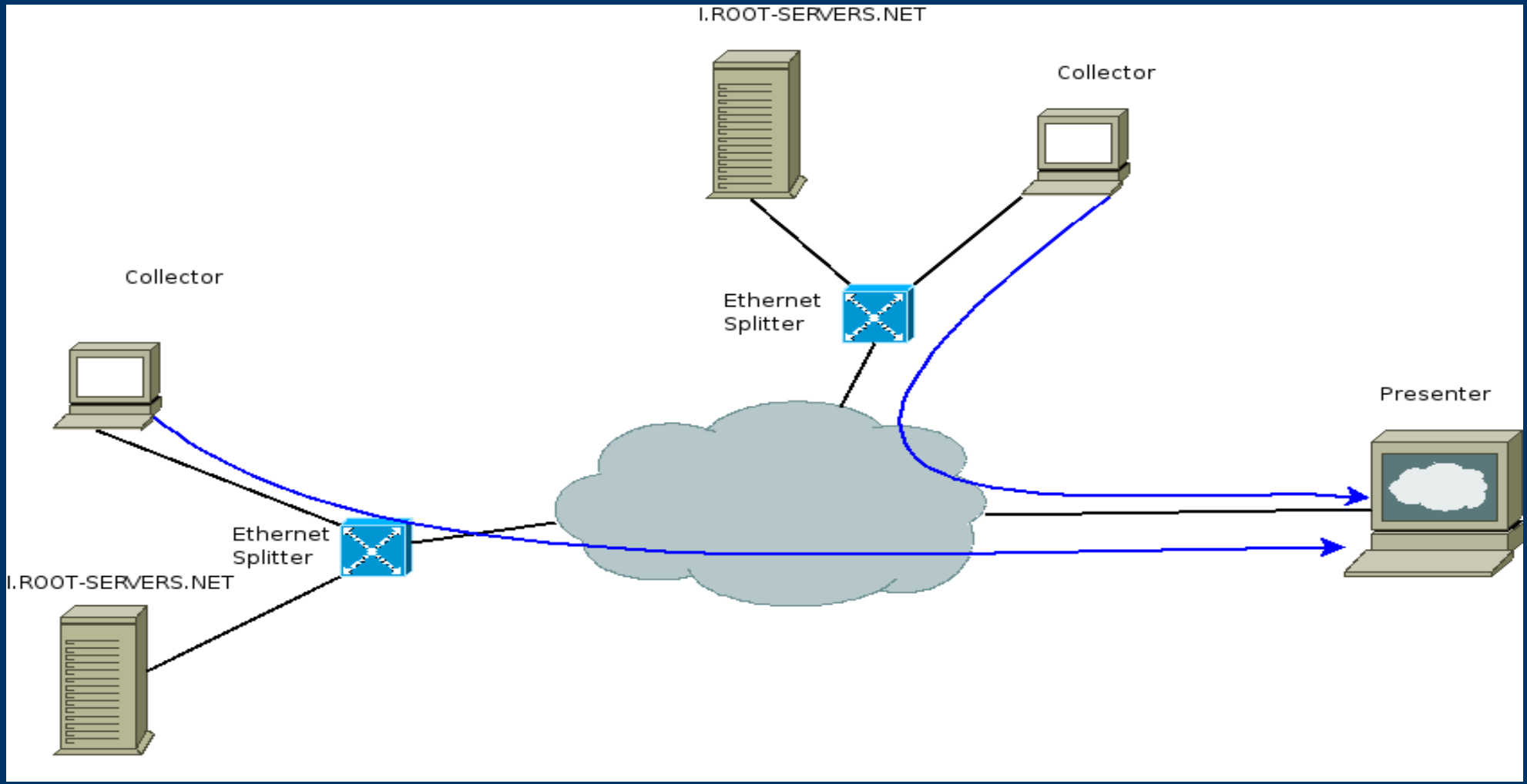
Where DSC comes in

- Mean to analyse traffic behaviour between different locations.
- Be able to compare traffic patterns (Time/location aso).
- Centralized data collection.
- Filter out unnecessary information.

Original DSC

- Collector
- Presenter
- Ethernet splitter
  - splits traffic to both
    - DNS
    - Collector

Orginal DSC


I.ROOT-SERVERS.NET
Collector
Collector
Ethernet
Splitter
Presenter
Ethernet
Splitter
I.ROOT-SERVERS.NET

Collector: brief overview

- Sniffs packets using pcap-lib.
- Collects statistics.
- Sends data to the collector using:
  - Https
  - Rsync/ssh
- Static configuration
- Writes data to Xml files before sending.

Presenter: Brief overview

- Saves data in plain files.
  - using directory hierarchies
  - reformats xml files to ploticus data files
- Uses ploticus for creating graphs.
- Programmed in Perl.

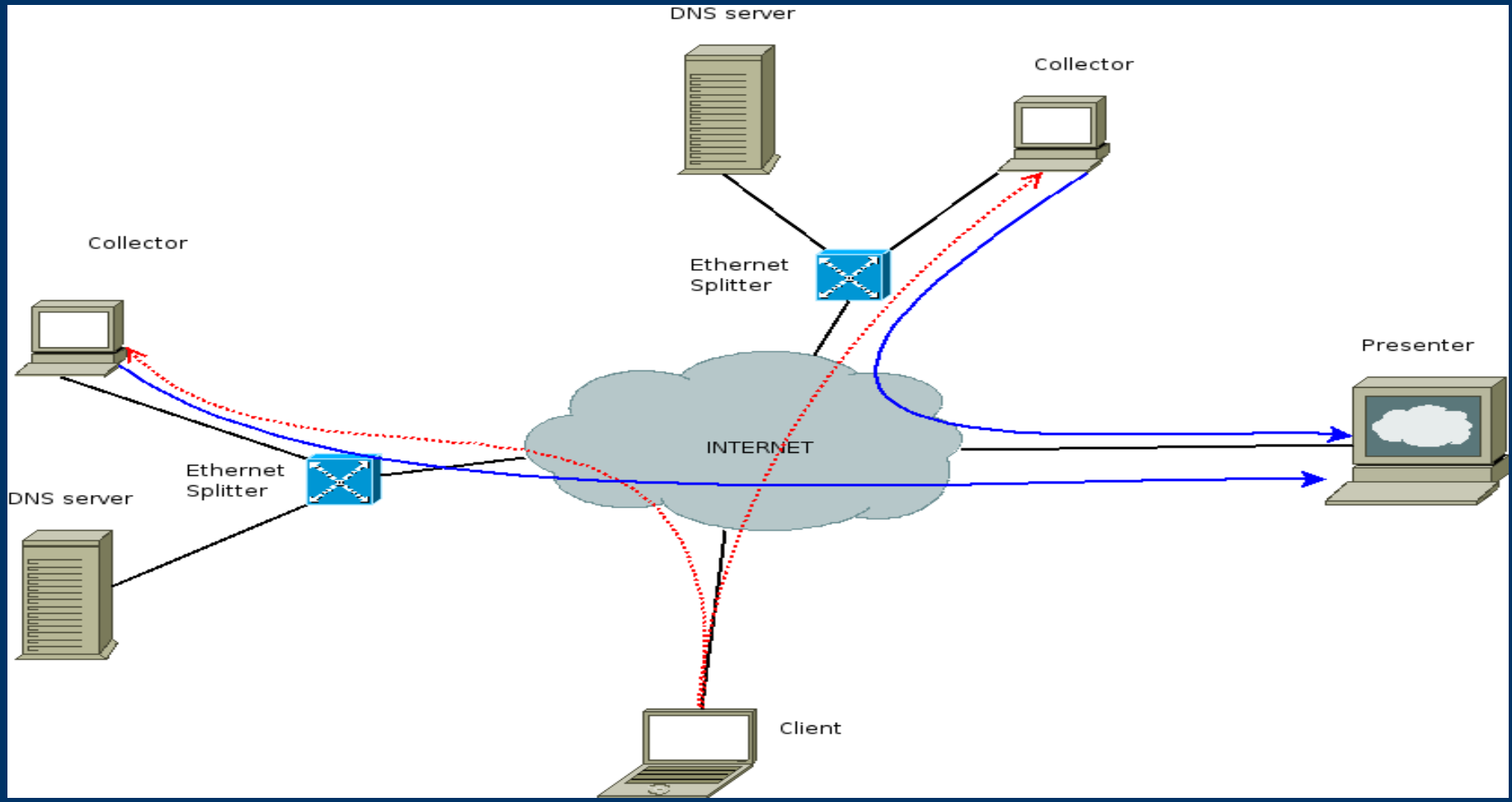
Autonomica Requirements

- To be able to remotely configure each collector.
  - For fast reconfiguration during traffic abnormalities
  - Collect specific data during a certain time
  - Be able to suspend collection
- More dynamic presentation.
  - Use known database for storing data
  - A more dynamic presentation
    - E.g. Compare already collected data

Basic overview of DSC-Mithril

- Configure collector remotely
  - During Runtime
    - Configure filters
    - Choose interface
    - Add datasets
    - .....
  - Safe and secure connection
    - Using a SSL tunnel

Basic overview of DSC-Mithril



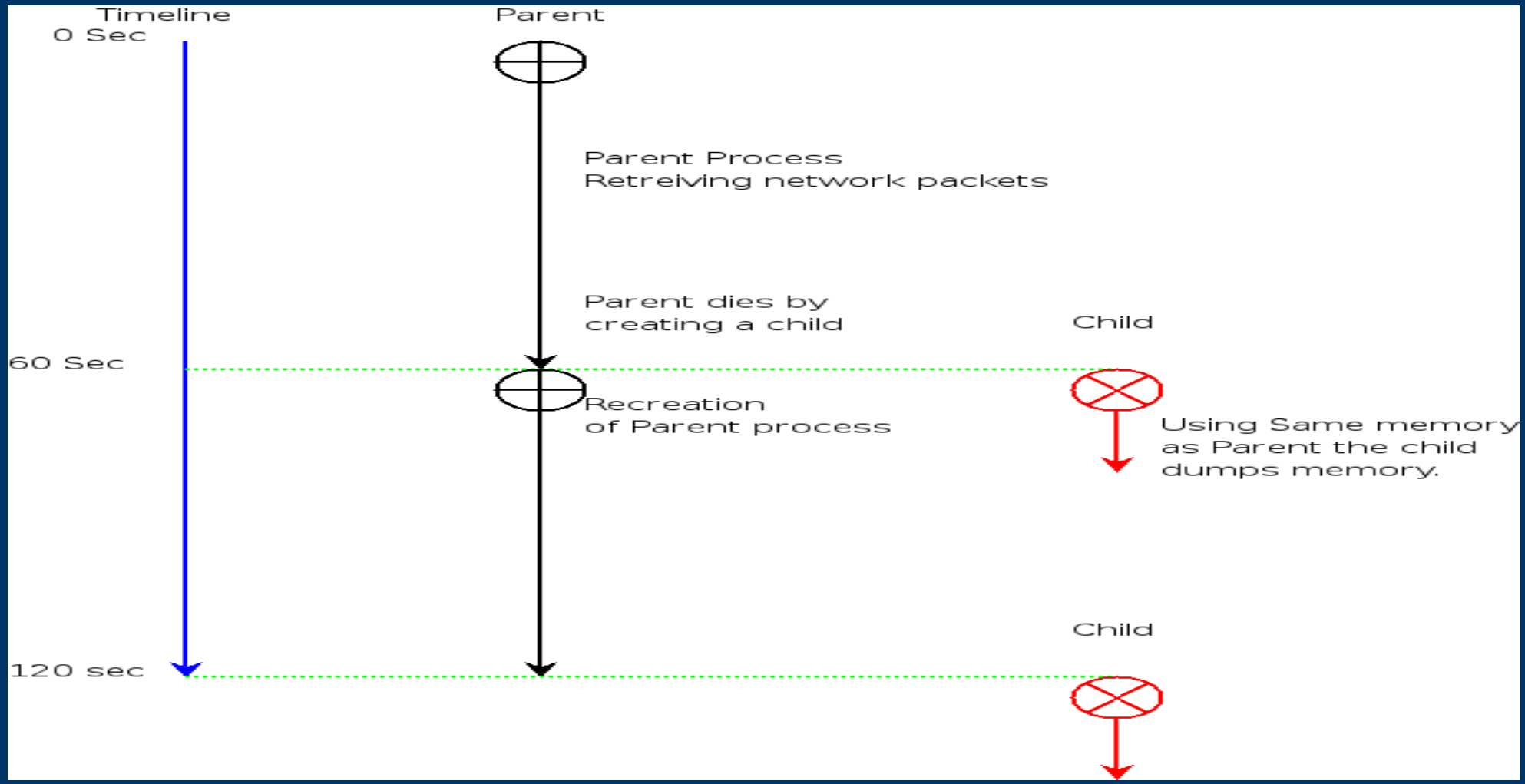
Changing DSC

- Collector design: From “Fork” to “Threads”
  - advantages:
    - Share memory
    - Process communication
    - Extendibility
  - Disadvantages:
    - Memory management (malloc/free)
      - A careful investigation of original DSC
    - Redesign of certain structures

DSC with FORK

- Parent process
  - 60 sec packet sniffing
  - create child process
  - restart
- Child Process
  - Same memory as parent
  - Writes to disk.
  - dies

DSC with FORK



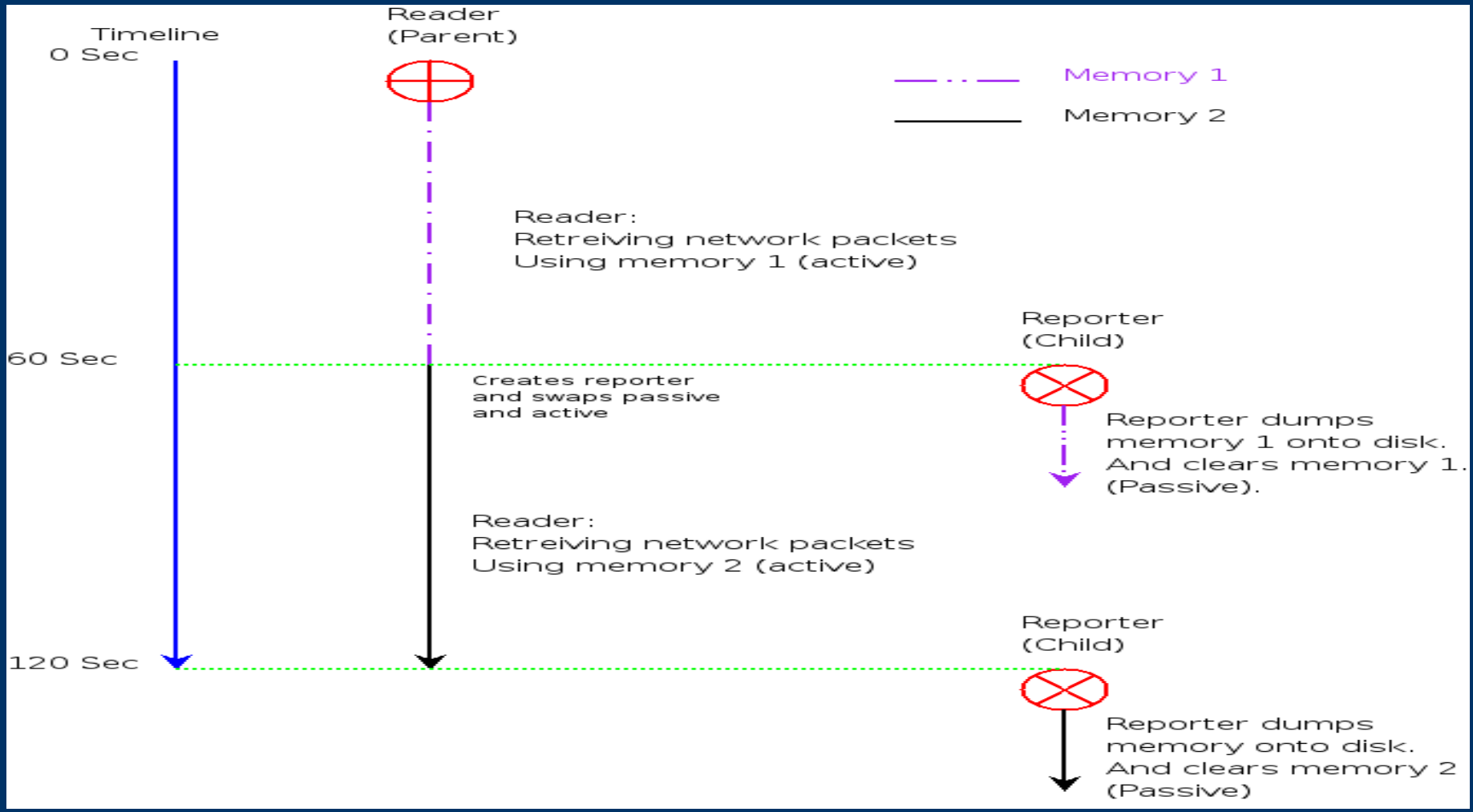
DSC with Threads

- A Reader thread
  - collects packets and updates statistic structure
  - Creates Reporter thread after 60 sec
  - Swaps active and passive memory structure
- A Reporter thread
  - Writes passive memory to disk
  - Clears passive memory

DSC memory management

- Same structure allocated twice
- Active and Passive memory
- Swap after 60 sec.
  - Active memory structure
    - Gets updated during packet collection by Reader thread
  - Passive memory structure
    - Read by reporter thread and dumped to disk
    - cleared after read

DSC Threads overview



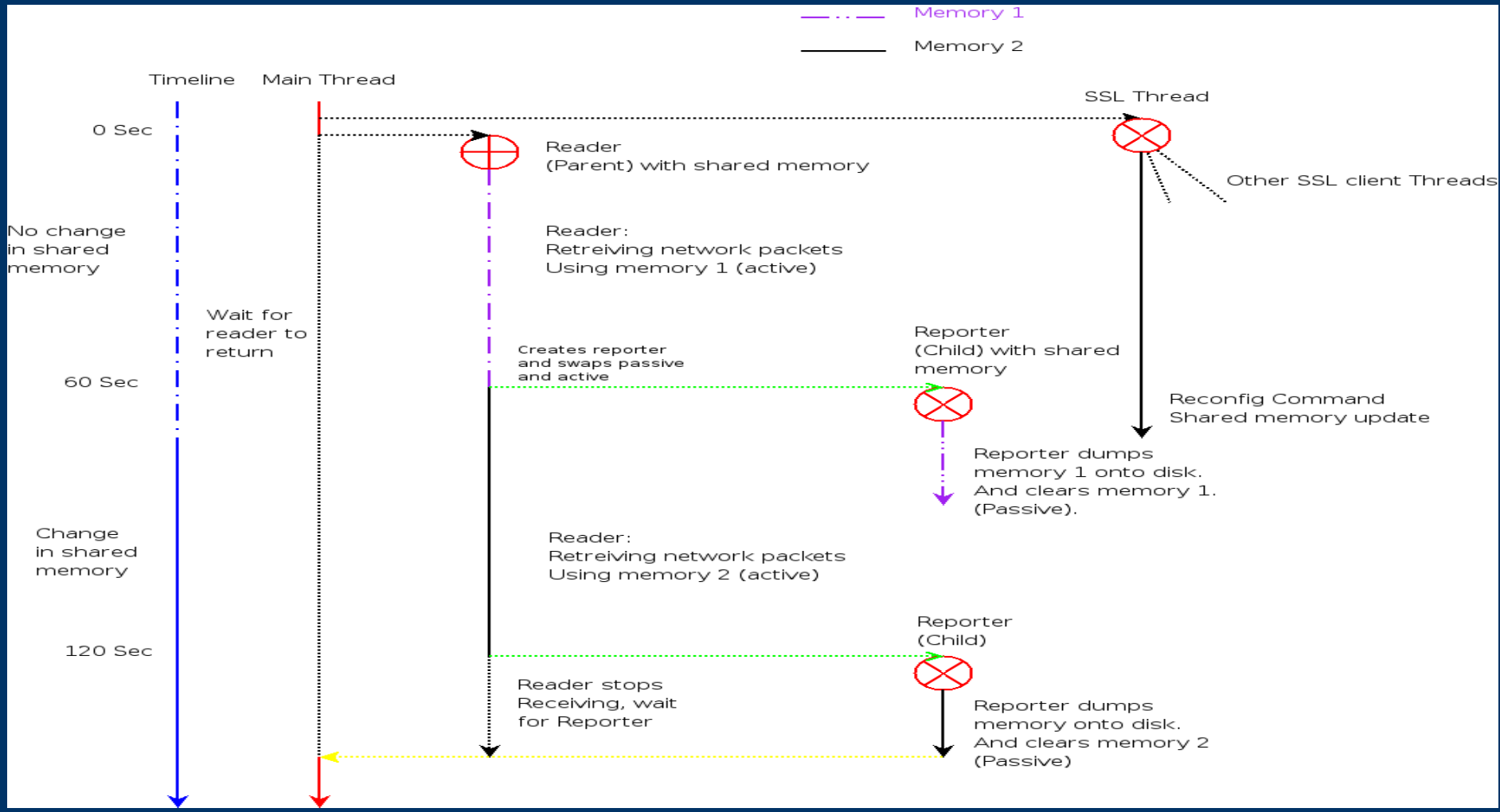
DSC Thread more....

- Configuration parser
  - TmfHapy could not parse twice in the same session
  - Replaced using BNF converter
    - BNFC uses Flex and Bison
      - Using labelled BNF grammar configuration
    - Creates a C parser

DSC with SSL Support

- Two new threads incorporated in the design
  - SSL Thread:
    - New Shared memory structure
      - Only SSL thread writes
      - All other threads reads
    - Connection based on certificates (x509_v3)
    - Multi threaded
  - Main Thread
    - Handles and schedules the other threads
    - Reads configuration
    - Initialization/destruction of memory structures.

DSC with SSL support



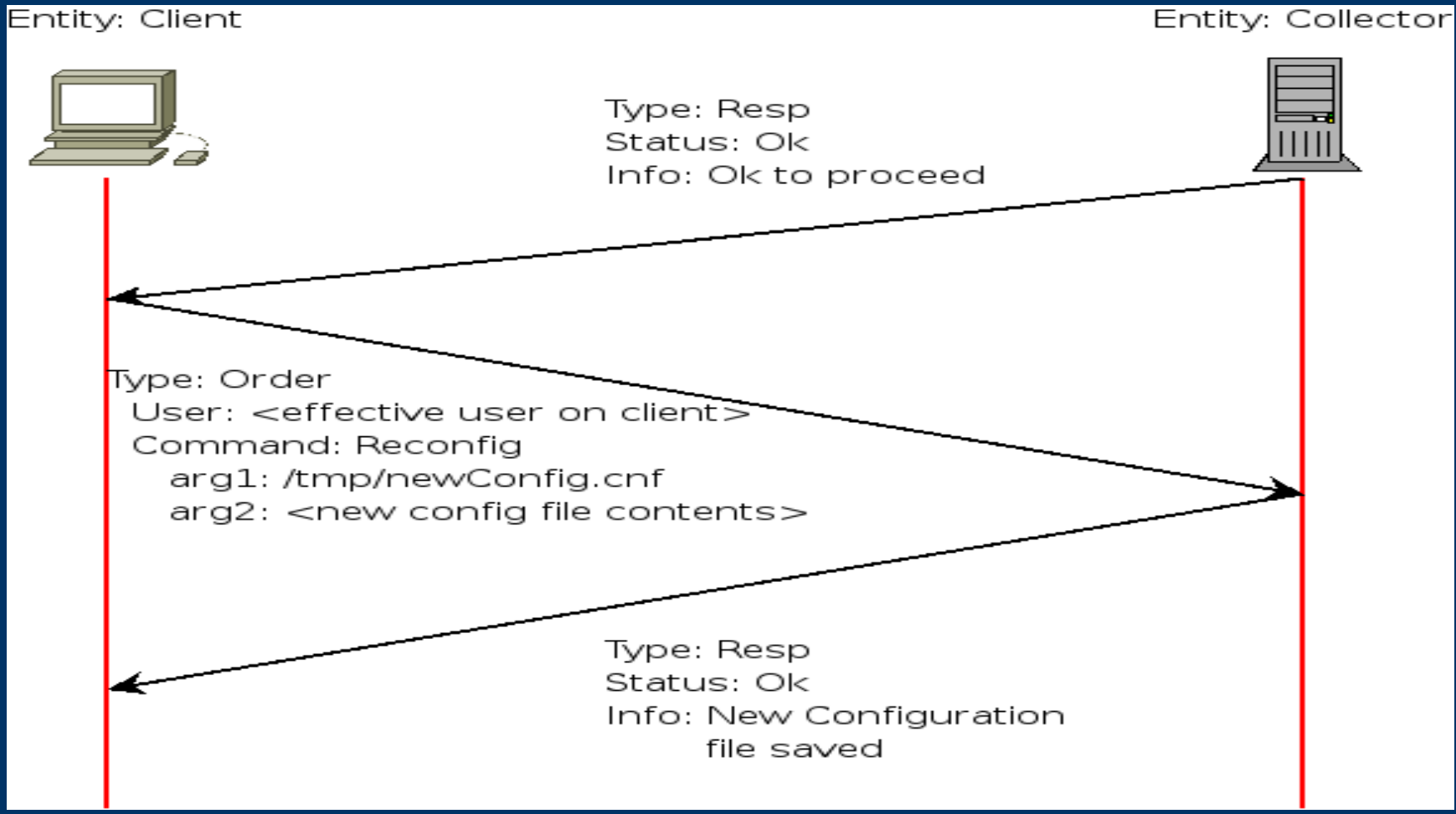
SSL authentication

- Certificate x509_v3:
  - Server
    - Needs same CA-chain that presented from client
    - Client SubjectAltName needs to be configured in server
    - Highly configurable using XML configuration file
    - Authorize client to perform action
  - Client
    - Present certificate to the server
      - needs to be signed by server-known CA

Underlying protocol

- Based on XML
  - Parser based on libxml.
  - Two types of messages
    - Order
      - User
      - command
      - Argument(s)
    - Response
      - Status
      - code
      - info
      - Data

Protocol Sequence



Typical Order Msg

- <Order>
  - <User> {Effective UID on Client}</User>
  - <Command> Reconfig </Command>
  - <Command_Info>
    - <Arg1> /tmp/newConfig.cnf</Arg1>
    - <Arg2>{Config file content} </Arg2>
  - </Command_Info>
- </Order>

Typical Response Msg

- <Resp>
  - <Status>
    - <Code> {Syslog code} </Code>
    - <Info> {Textual information} </Info>
  - </Status>
  - <Resp_Data>
    - {data recevied from response}
  - </Resp_Data>
- </Resp>

Future development

- Presenter
  - Save data into database instead (SQL)
- Collector
  - Testing!!
  - More functionality

Thank you!

- Thats all...
- By the way, DSC-mithril will be open for public