C ontrol system based on a
H ighly
A bstracted and
O pen
S tructure

software architecture and developer introduction

C. Bisegni

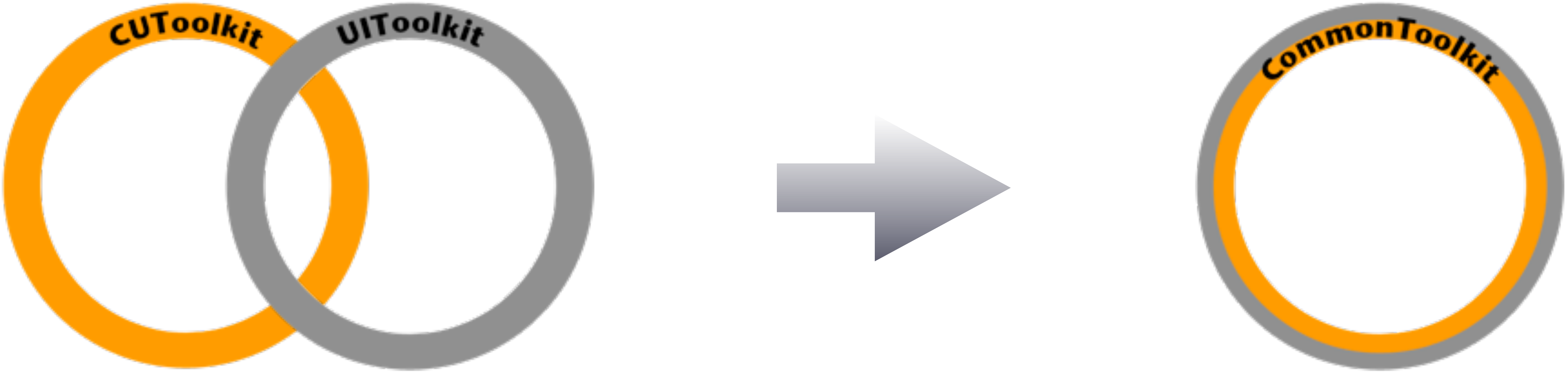
!CHAOS software layer

!CHAOS software layer

CUToolkit, abstract the !CHAOS resources to the device drivers developers.

UIToolkit tools for developing client application that accesses !CHAOS resource

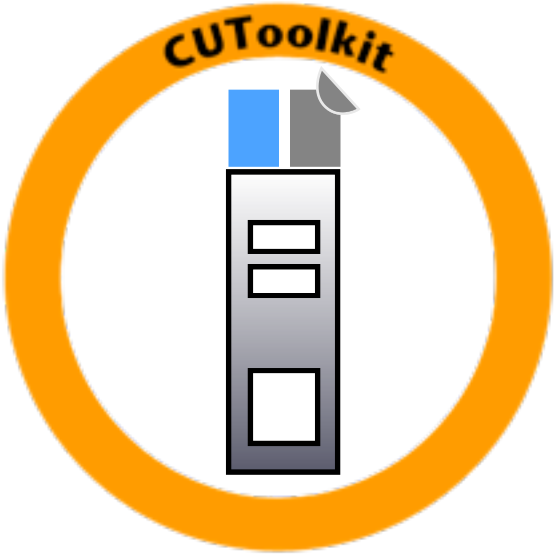
!CHAOS software layer



- The two layer are based on **CommonToolkit** and all they are the CHAOS Framework
- Developed in c++
- Multi Threading

!CHAOS Node & Service

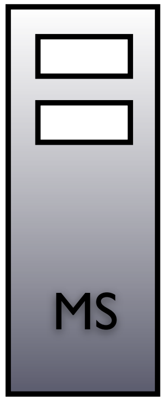
!CHAOS FrontEnd and User Node



Control Unit, a piece of software developed on CUToolkit implementing the device drivers

The Chaos Control GUI is based on the UIToolkit for accessing !CHAOS resources. The UIToolkit is also used by control panels/client applications developers to make their custom application

!CHAOS Middle Layer



MetaData Server, keep track of all information about device DataSet and Command, CU address and other info.

MEMCACHED

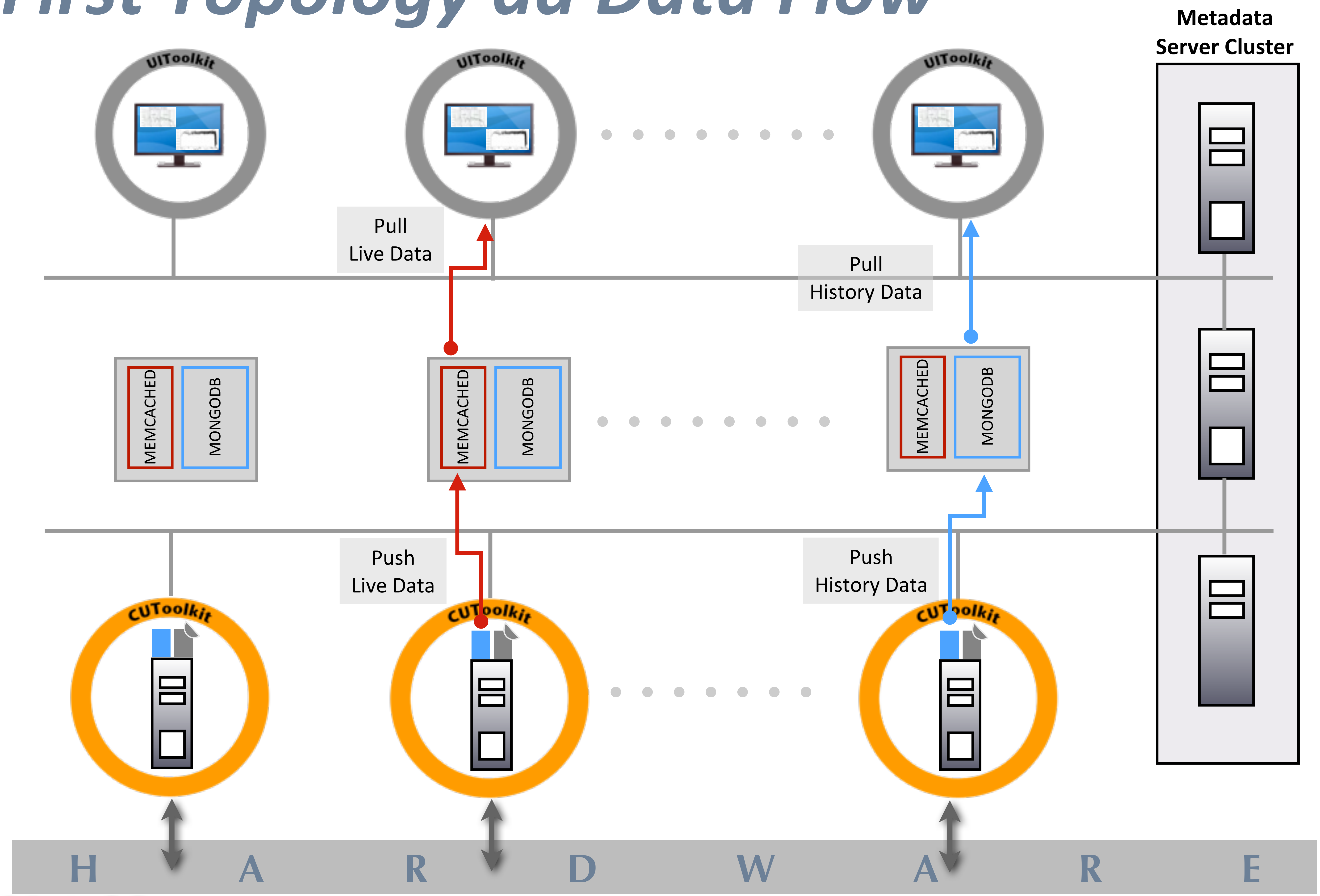
Memcached is used for caching lived data

MONGODB

MongoDB is used for storing history data

Topology and Data flow

First Topology ad Data Flow



Topology Next Step

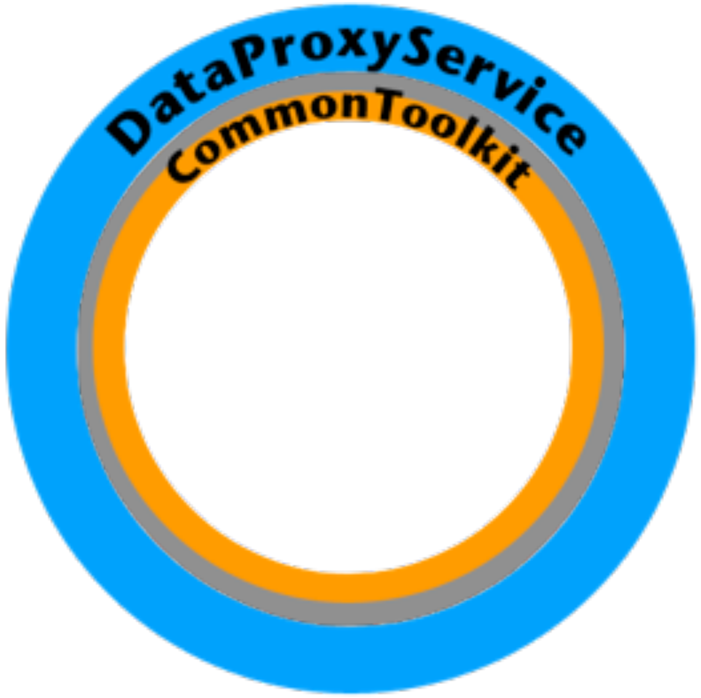
We are in R&D so we have made some adjustment to the CHAOS topology

history data must be managed in different way for different kind of query:

- **near time history data**
- **long time history data**
- **data warehouse query**
- **etc.**

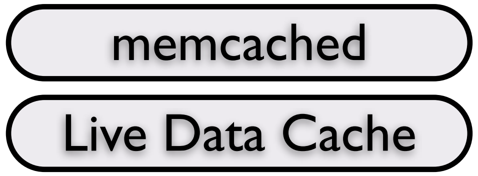
New Node & Service

!CHAOS Client & MS Node



Data Proxy Service, is a scalable service that implement a common proxy for the Live and History data services. It includes memcached and the drivers for implementing ChaosQL for storing or querying history data

!CHAOS Service


memcached

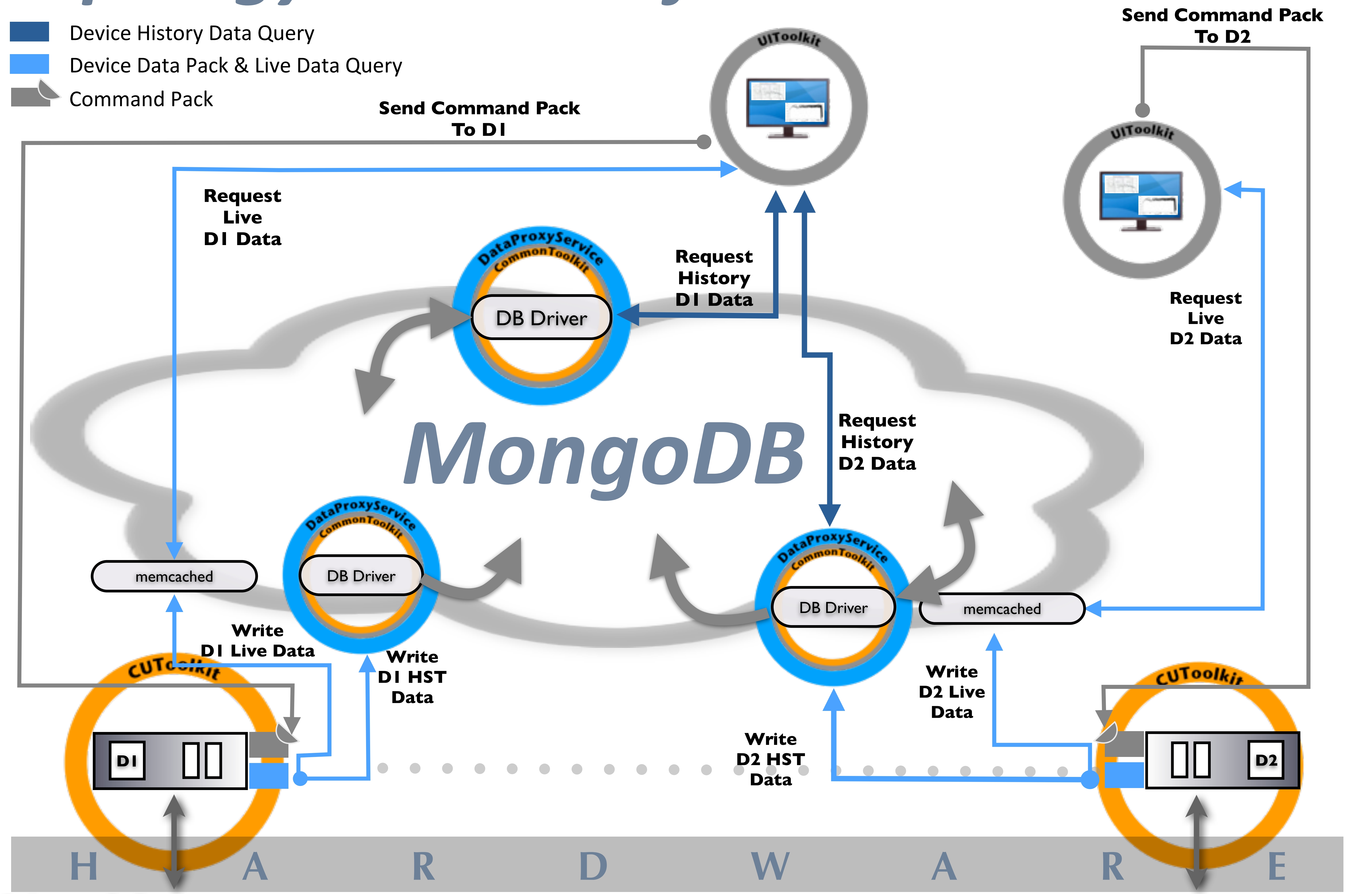
Live Data Cache

Live data cache is a service implemented with Memcached

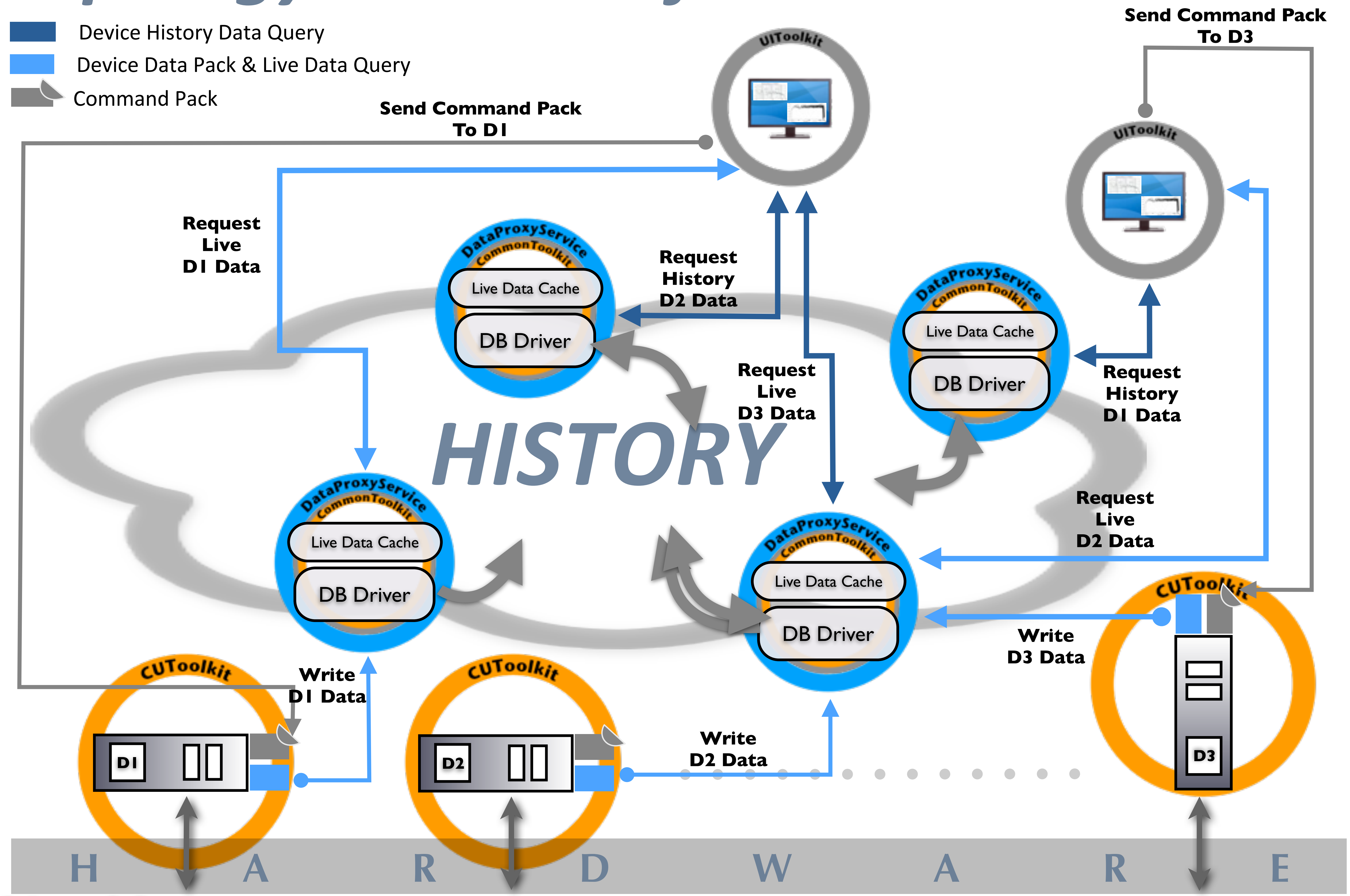
This is the History Storage Cloud, that can be accessed by means of the Data Service Proxy.

New Topology and Data flow

Topology and Data flow



Topology and Data flow



Topology Next Step

Scaling with *Memcached* and *MongoDB*

Memcached is a key-value cache and scales on key names, each client has an algorithm to link a “key” with “server”

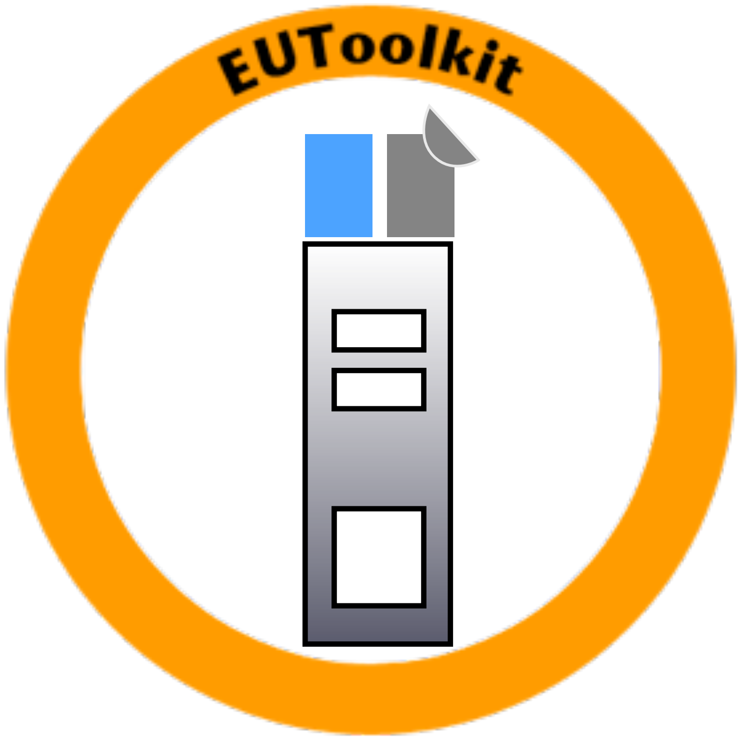
MongoDB scales well on writes and reads (some tricks may be used to increase the write speed)

New Idea for the Control Automation and Computing

An idea for automate the controls

- we are trying to design and add a new “node” into CHAOS
- it will be like a Control Unit but modelled to be used only for control other CU or make computing
- it will be used for create a distributed final state machine
- We called it Execution Unit

Execution Unit Node



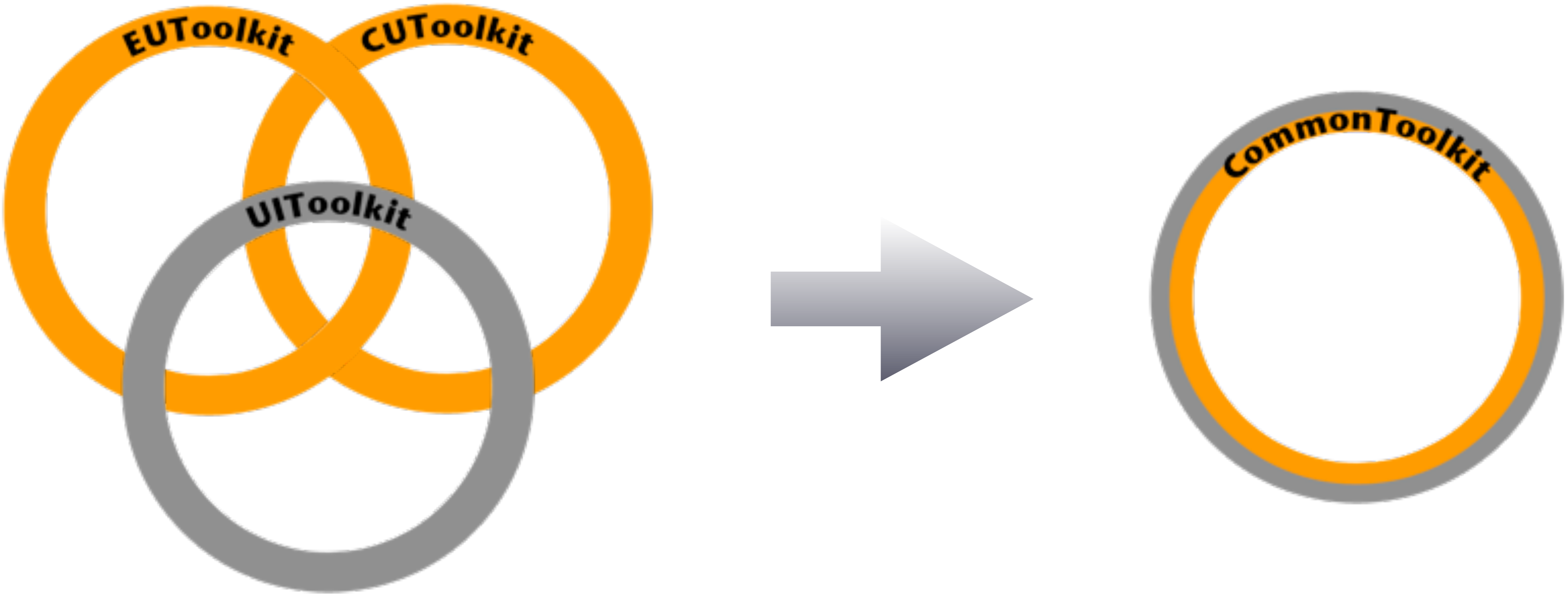
Execution Unit, is a the specialized software that implement control or computing algorithm

🟦 ChaosQL Data Pack Channel

⬛ Chaos Command Pack Channel

ExecutionUnit must define the input and output class of data(HW Dataset or Basic Element) that are needed to do the work

!CHAOS framework with new layer

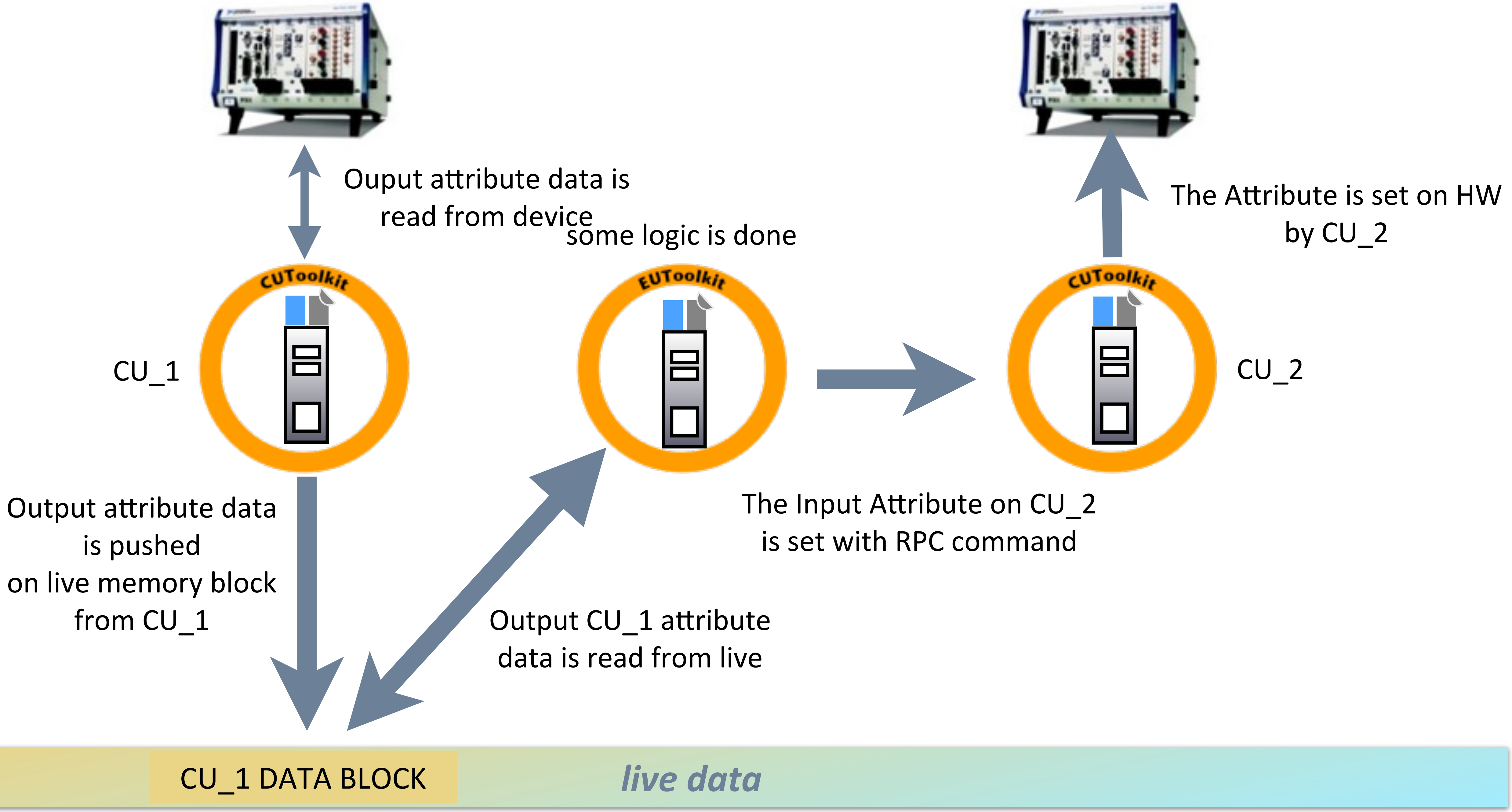


CUToolkit (as for previous Layer) is implemented on CommonToolkit

ExecutionUnit in the work

Execution Unit Example 1

this is the real data flow



Ouput attribute data is
read from device

some logic is done

The Attribute is set on HW
by CU_2

CU_1

CU_2

Output attribute data
is pushed
on live memory block
from CU_1

The Input Attribute on CU_2
is set with RPC command

Output CU_1 attribute
data is read from live

CU_1 DATA BLOCK

live data

!CHAOS compared to a Normal PC

- CHAOS can be considered like a distributed computer:
  - Live data is the RAM
  - History data is the Hard Disk
  - CU are the kernel driver
  - EU are the process that do something

Common Toolkit

Common Toolkit

- CommonToolkit has tree important software layer
  - BSON Container for hardware dataset abstraction and RPC pack
  - RPC Driver
  - ChaosQL Driver

- in addition it has a lot of common utility code

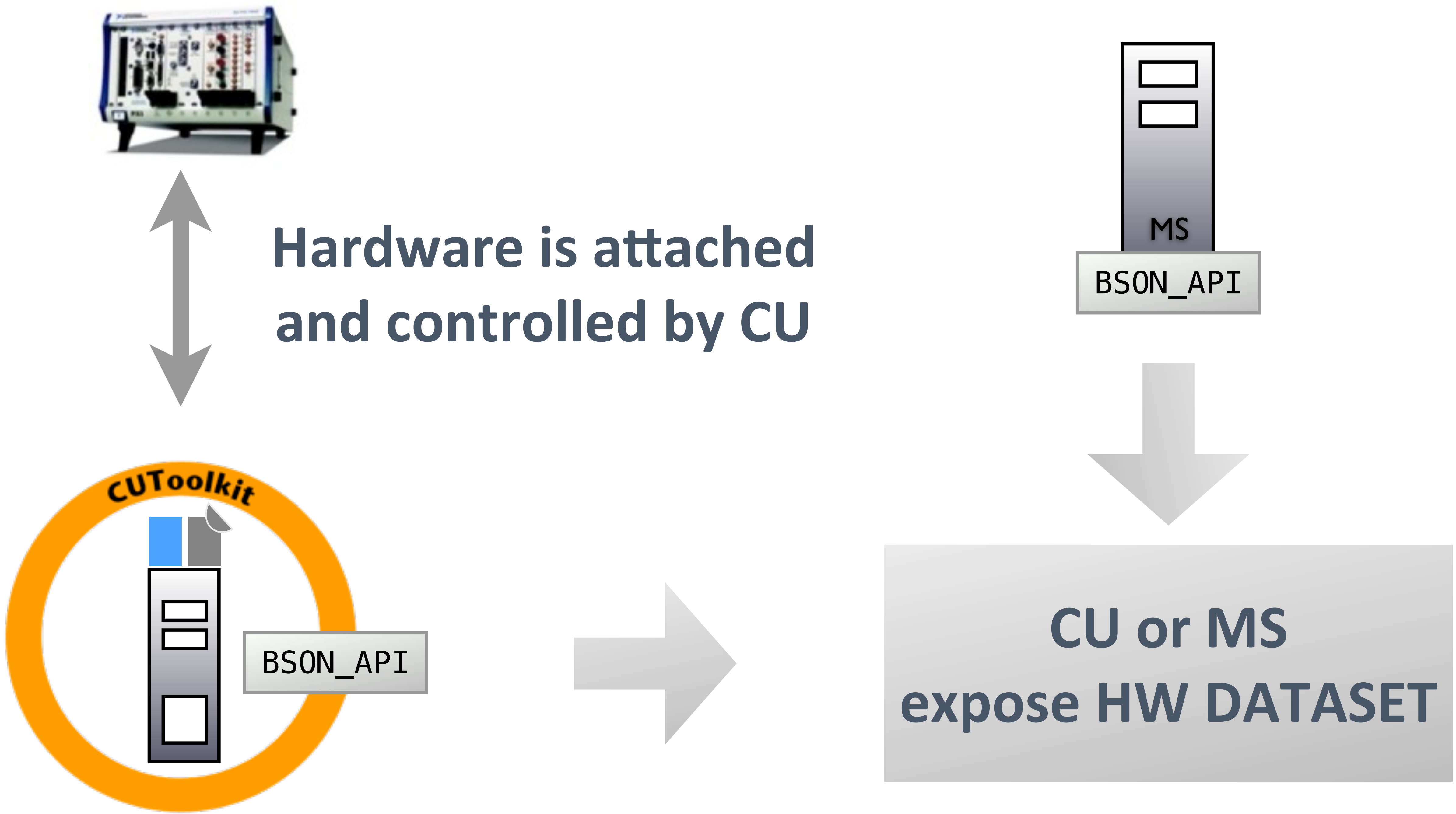
CommonToolkit

abstraction

abstraction

- CHAOS use BSON(http://bsonspec.org/) for data description and serialization
- Binary JSON like document
- it is used in:
    - Hardware attribute description
    - RPC message between node

Hardware abstraction



Hardware is attached and controlled by CU

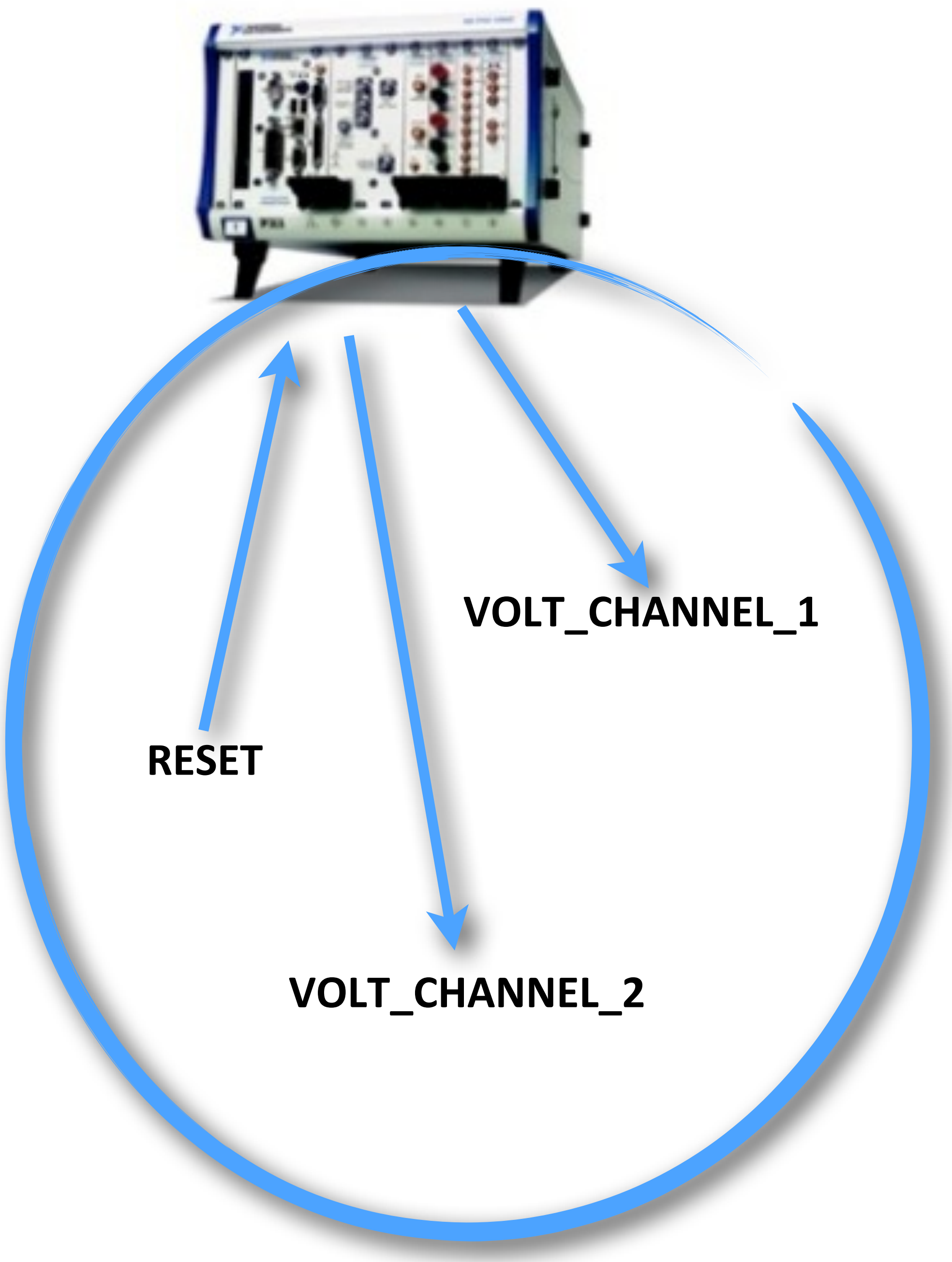
CU or MS expose HW DATASET

Hardware DATASET

Hardware DATASET

- Hardware attribute are described within DATASET
- Each attribute is defined by
    - Name
    - Description
    - Type (kind+basic type ex: VOLT32, CUSTOM_STR, etc...)
    - Cardinality (single or array)
    - Flow (Input, Output, Bidirectional)
    - Range

Hardware DATASET



DATASET

name:VOLT_CHANNEL_1
type:VOLT32
flow:output
range:1-20
card: 1

name:VOLT_CHANNEL_2
type:VOLT32
flow:output
range:1-20
card: 1

name: RESET
type: BYTE
flow: input
card: 1

CommonToolkit

RPC System

RPC System

- *RPC System is implemented as a plug-ins System.*
- *It's is abstracted to internal CHAOS framework so we con change it*
- *The RPC layer is only used to send and receive bson data*
- *all the information are in the BSON pack*

RPC System

- The RPC System is user for:
    - MS <-> CU
        - CU management and retrieval information
        - CU Heartbeat
    - UI <-> CU
        - Set the input attributes of hardware dataset
        - CU management and retrieval information
- CU Management is:
    - init, deinit, start and stop

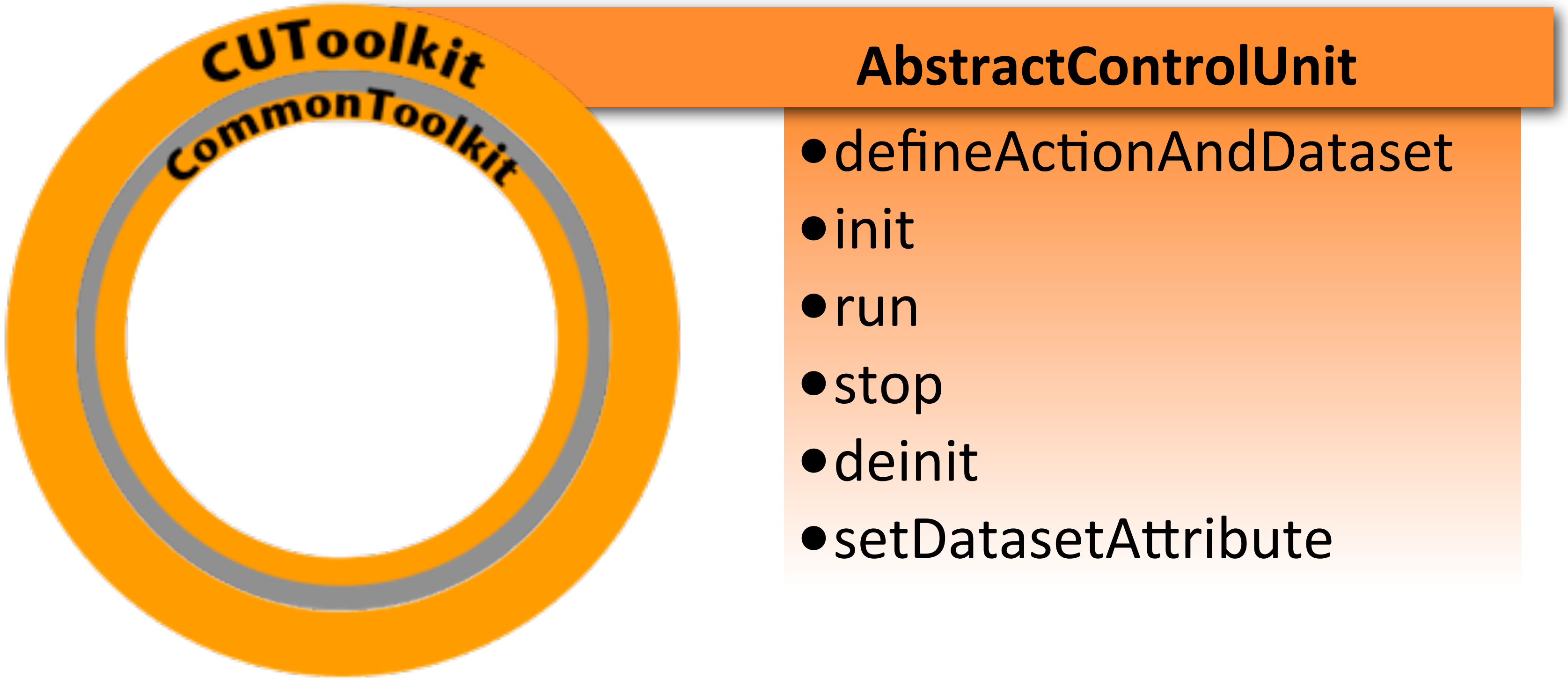
CUToolkit

CUToolkit and Control Unit

- CUToolkit help the Control Unit development
- Developer need to extend only one class to create a CHAOS driver, the “AbstractControlUnit”
- AbstractControlUnit expose all the necessary api for interact with CHOAS

Hardware controller (CU)

AbstractControlUnit is an abstract cpp class, that force developer to implement some method



AbstractControlUnit

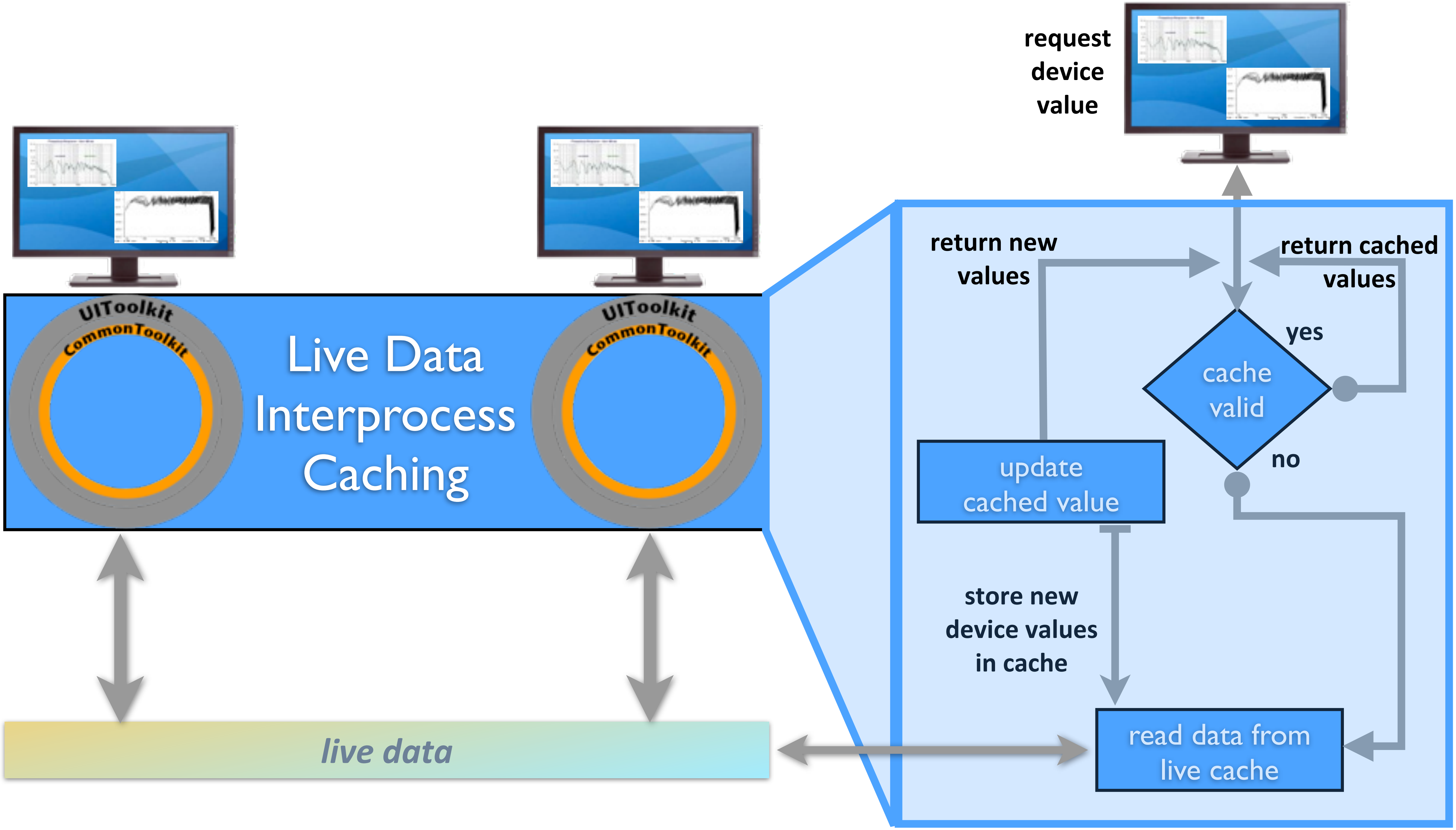
- defineActionAndDataset
- init
- run
- stop
- deinit
- setDatasetAttribute

UIToolkit

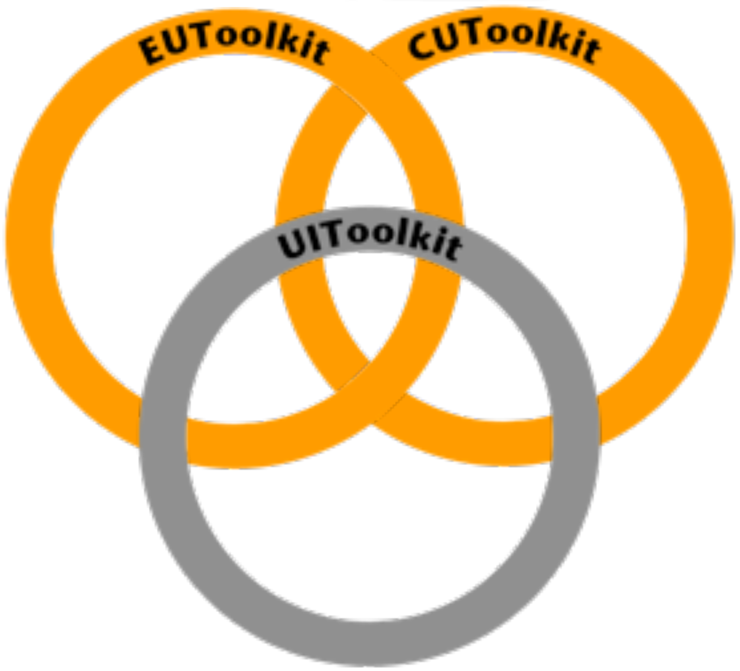
UIToolkit and unified Control GUI

- UIToolkit is the framework layer that permit to developer to create client application that need to access CHAOS resource
- it abstract to application:
    - connection to CU for control a device
    - querying the MetadataServer for retrieve HW information and Dataset
    - caching across UIToolkit process for live data
    - intelligent polling(predict when there will be a new valued on live data storage)
    - Other functionality are in study

UIToolkit and unified Control GUI



!CHAOS



thanks for the time