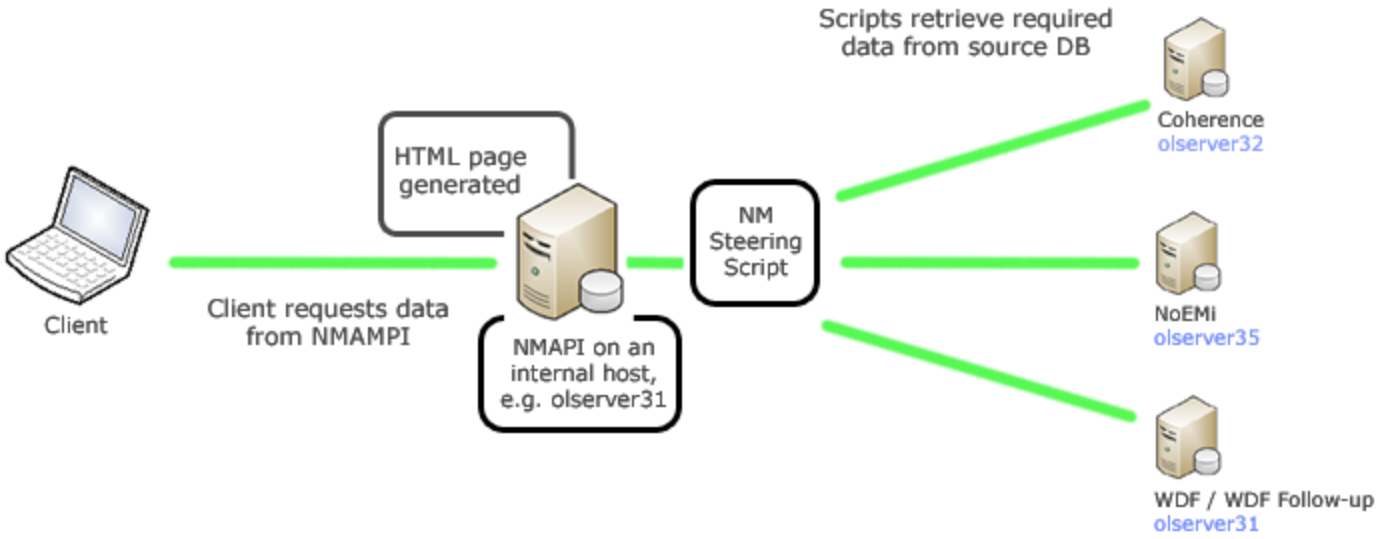
NoEMI@Caltech

- Interest in running NoEMI tool on LIGO data has been raised by LIGO people
- NoEMI is one of the Noise Monitors tools and is integrated in the NMAPI framework.
- We take the opportunity and propose to work on a **common interface** for Noise Monitors for the Advanced Detectors.

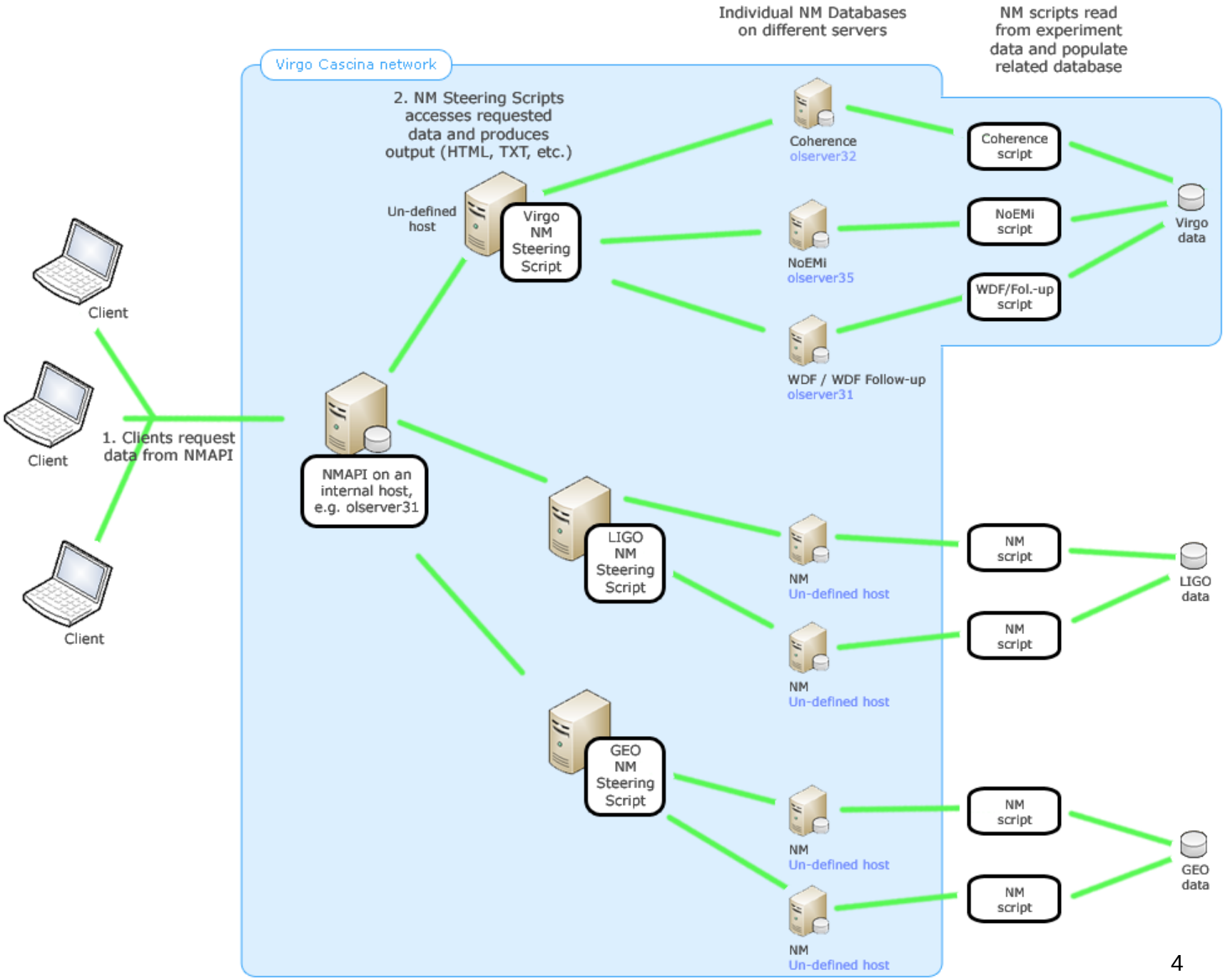
NMAPI-Architecture

- NMAPI currently sits on one single pc.
- It sits and waits for requests, passed via HTTPS.
- Upon request receipt, the information is passed to the dedicated Noise Monitor *Steering Script*, which deals with retrieving requested results from data source.

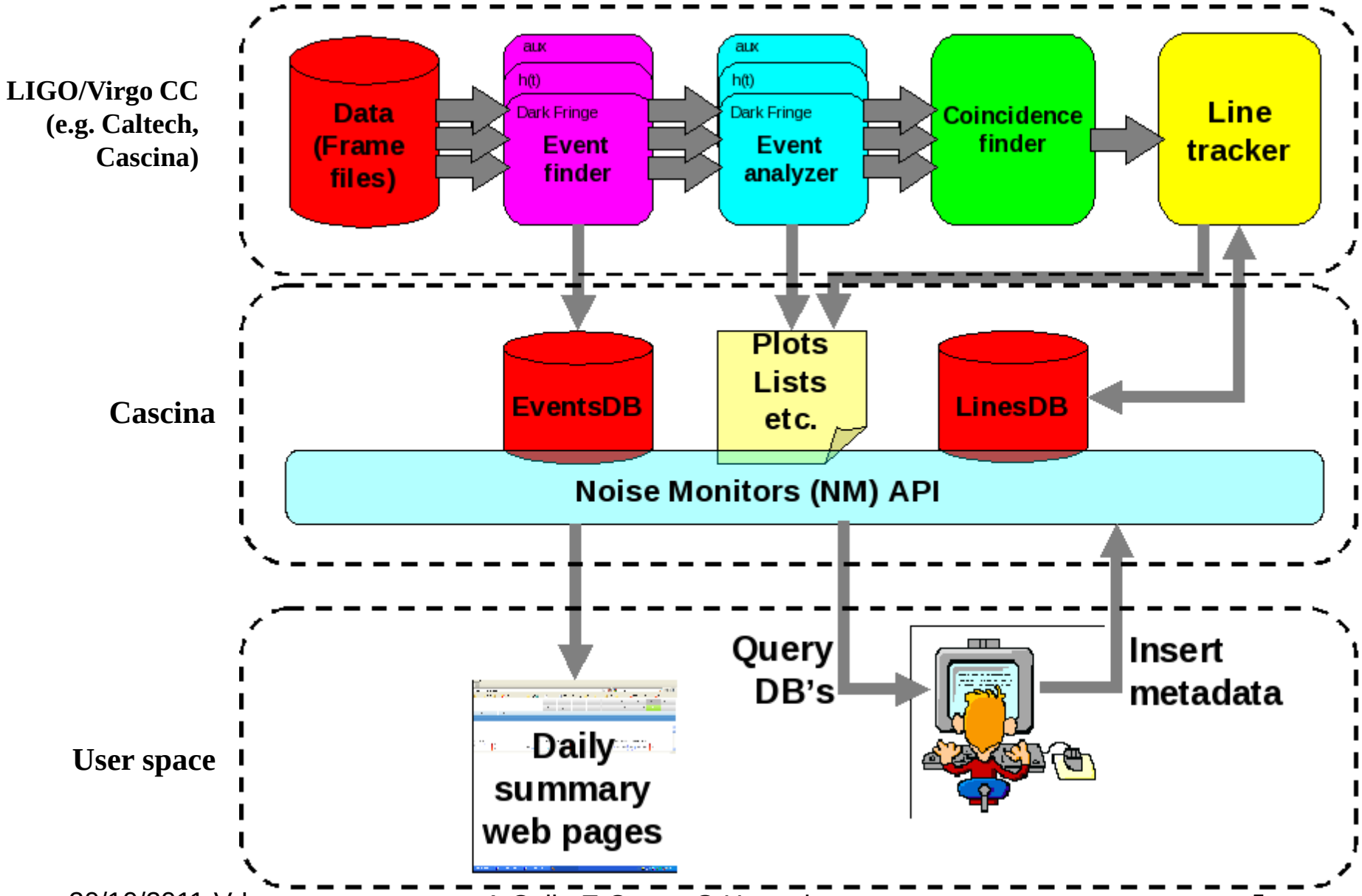


Distributed-NMAPI Architecture

- We propose to make NMAPI working on more detectors data.
- Since the computational needs will be higher, we think about a distributed architecture
- The Web interface will be the same for all the detectors
- The framework is extensible thanks to its “plug-in” architecture: new NMs can be added in the future


Individual NM Databases on different servers
NM scripts read from experiment data and populate related database
Virgo Cascina network
2. NM Steering Scripts accesses requested data and produces output (HTML, TXT, etc.)
Un-defined host
Virgo NM Steering Script
Coherence olserver32
NoEMi olserver35
WDF / WDF Follow-up olserver31
Coherence script
NoEMi script
WDF/Fol.-up script
Virgo data
Client
Client
Client
1. Clients request data from NMAPI
NMAPI on an internal host, e.g. olserver31
LIGO NM Steering Script
GEO NM Steering Script
NM Un-defined host
NM Un-defined host
NM Un-defined host
NM Un-defined host
NM script
NM script
NM script
NM script
LIGO data
GEO data

NoEMi workflow (Test setup)



Test setup

- As a first check of the feasibility of our proposal, we plan to run NoEMi on the LIGO data, in a LIGO computing center (e.g. Caltech)
- Test will consist in analyzing a period of LIGO data, extract the noise lines and fill the NM database at Cascina

Software requirements

- Analysis of 1 single channel requires < 2 GB RAM and one CPU
  - Data (frame files) must be accessible locally
  - e.g. at Cascina data disks are NFS mounted
- Noise Monitor & database server available at Cascina
- Python 2.5.2 + some modules (matplotlib, numpy, MySQLdb, markup)
- MySQL port open for incoming & outgoing traffic
- Possibility to send locally created plots & web pages to web server

Storage requirements

What	Disk size / 1 day	Disk size / 1 year
Log Files - DF and auxiliary channels (1) (2)	100 MB	40 GB
Log Files – h(t) (3)	700 MB	260 GB
Log files - DF + h(t) + 27 auxiliary channels (1)	3.5 GB	1.3 TB
Events database - DF and auxiliary channels	50 MB	20 GB
Events database - h(t)	200 MB	70 GB
EVF database - DF+h(t)+27 auxiliary channels	1.6 GB	600 GB
Lists of lines & plots for the daily pages	0.4 GB	150 GB
Lines database	2 MB	0.7 GB
Plots for Lines table web page (4)	100 MB	100 MB
TOTAL		**2 TB**

(1) Log files can be deleted/zipped after the analysis

(2) the sizes refer to the 20 KHz sampled channels; the channels sampled at lower frequencies are smaller.

(3) h(t) analysis is done with a lower CR thresholds, therefore its size is bigger

(4) Lines table is rewritten everyday and its size is practically constant