DGAS

Today and tomorrow

andrea.guarise@to.infn.it

Workshop CCR – INFNGrid
17-21 Maggio 2010 – Santa Tecla

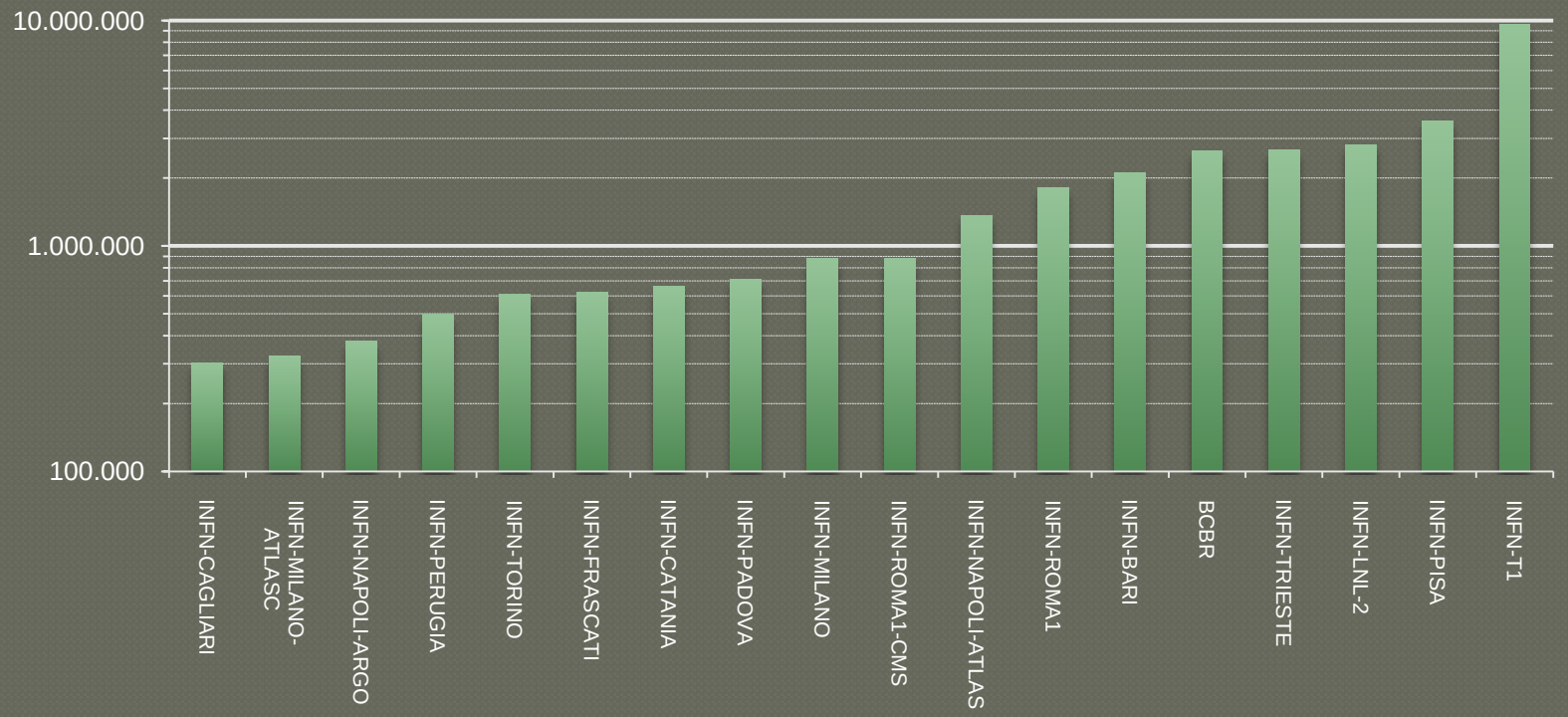
Today

Utilization

- DGAS is the accounting toolkit adopted in INFNGrid.
- The German grid initiative D-Grid is introducing DGAS in production, adpoting it's latest stable release (3.4.0).
- *HellasGrid* deployed an evaluation testbed for DGAS+hlrMon and is planning to use it in production.

Some numbers…

- Nearly 80 sites collects information for 18 millions record per year (and counting).
- Since version 3.4.0 the throughput, in the CE -> HLR record transfer , can be up to 100k record / day.



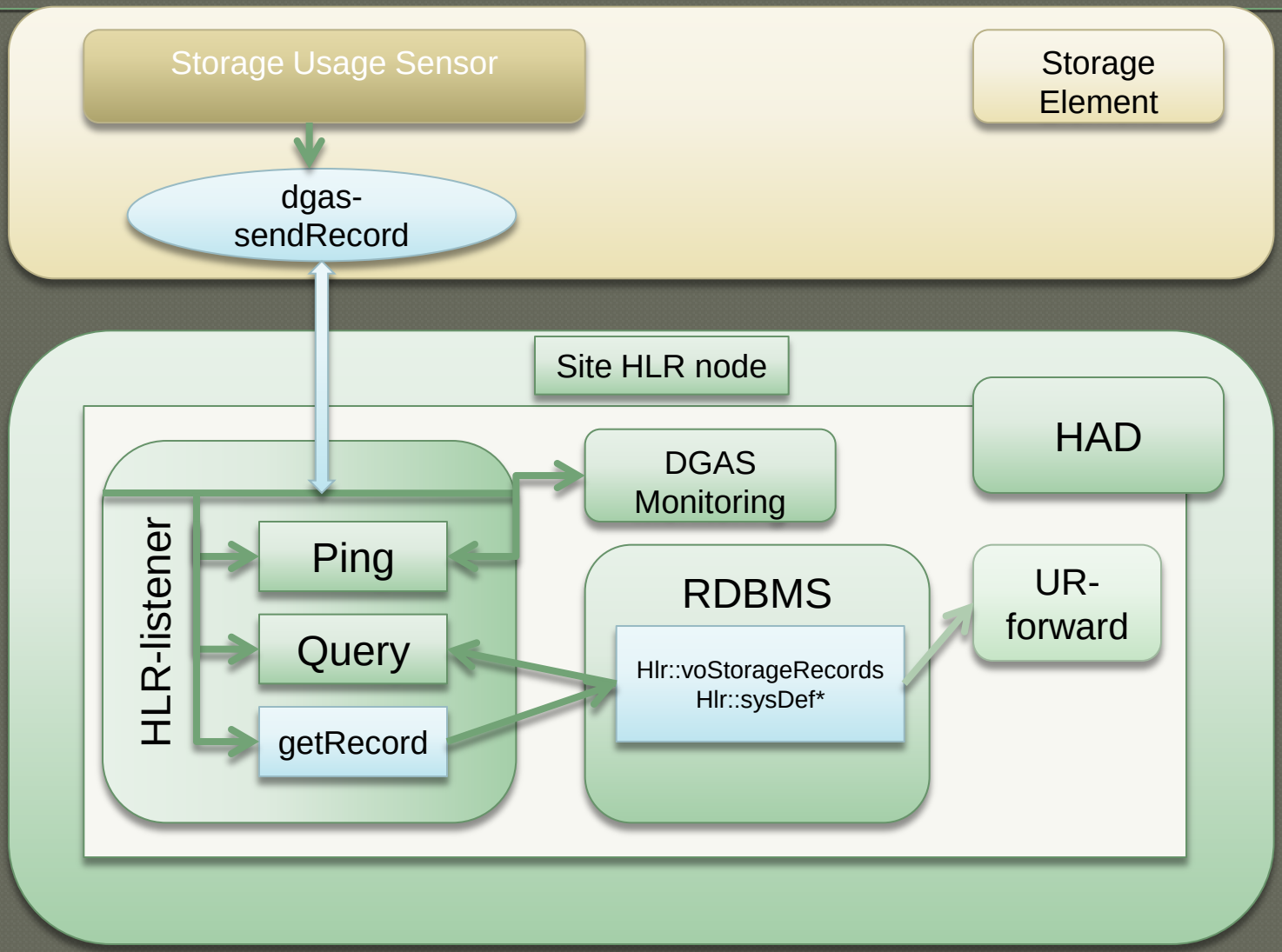
Release 3.4.0

- Since the end of 2009 a new major release is being deployed, introducing many new functionalities, such as:
  - VO-based 2lHLR, superseding the old user HLR,
  - New, simplified, database schema,
  - Faster client-server communication and improved throughput,
  - New record anti-duplication check,
  - DGAS2GOC integrated into DGAS distribution,
  - Free format benchmark types in DB schema,
  - System configurable translation rules (e.g. to translate VO such as glast into glast.org),
  - Added support for Condor (and experimentally SGE) batch managers,
  - VO mapping from pool account UID configurable via regexp rules,
  - Sensors can produce accounting records just for some of the VO,
  - Storage accounting toolkit,
  - Worker Node hostname in the Usage Record.
- This latest release is a major re-engineering of the software, and it's really young. still requiring some tuning effort.

Storage accounting

- DGAS provides a storage accounting functionality which enables sites to keep track of VO space consumption on their storage elements.
- Due to the highly heterogeneous environment in which Storage Accounting has to be operated, DGAS just provides a toolkit that sites can leverage to implement their accounting sensors. The whole storage accounting toolkit has been designed to be as simple as possible and highly customizable.
- Plugins, that should be provided by Site Managers, can be produced for all the storage elements currently used in production.

Storage accounting workflow

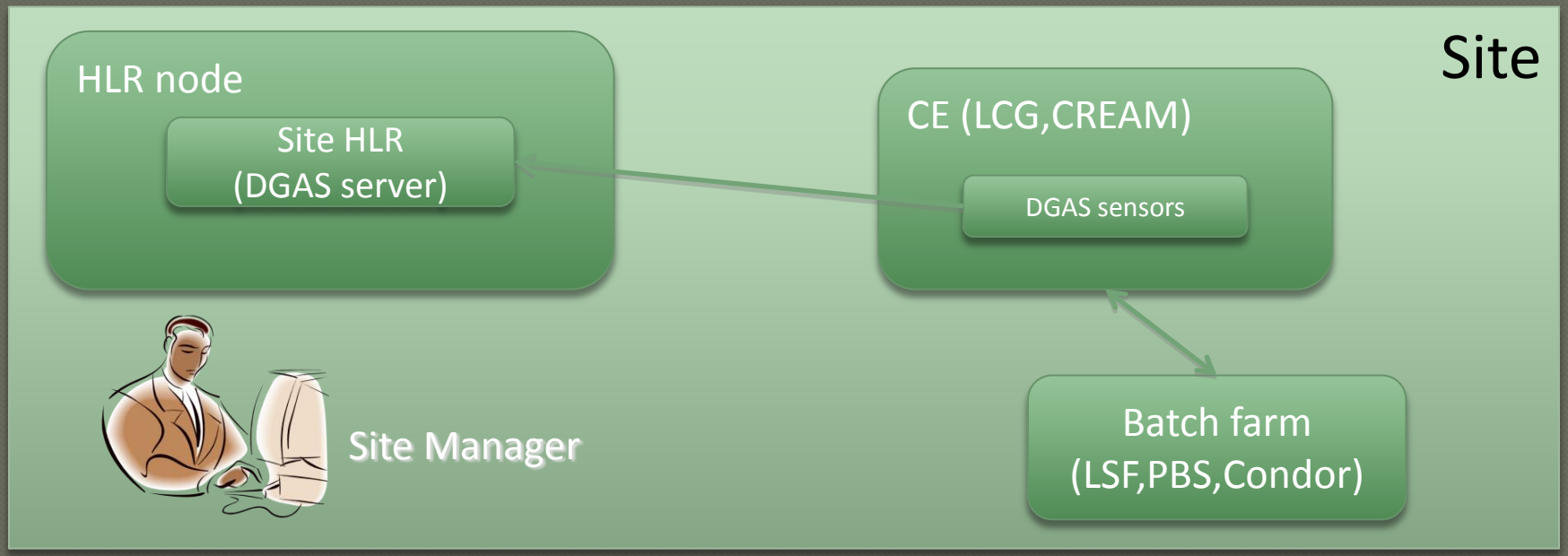


Tomorrow

Requests to be addressed

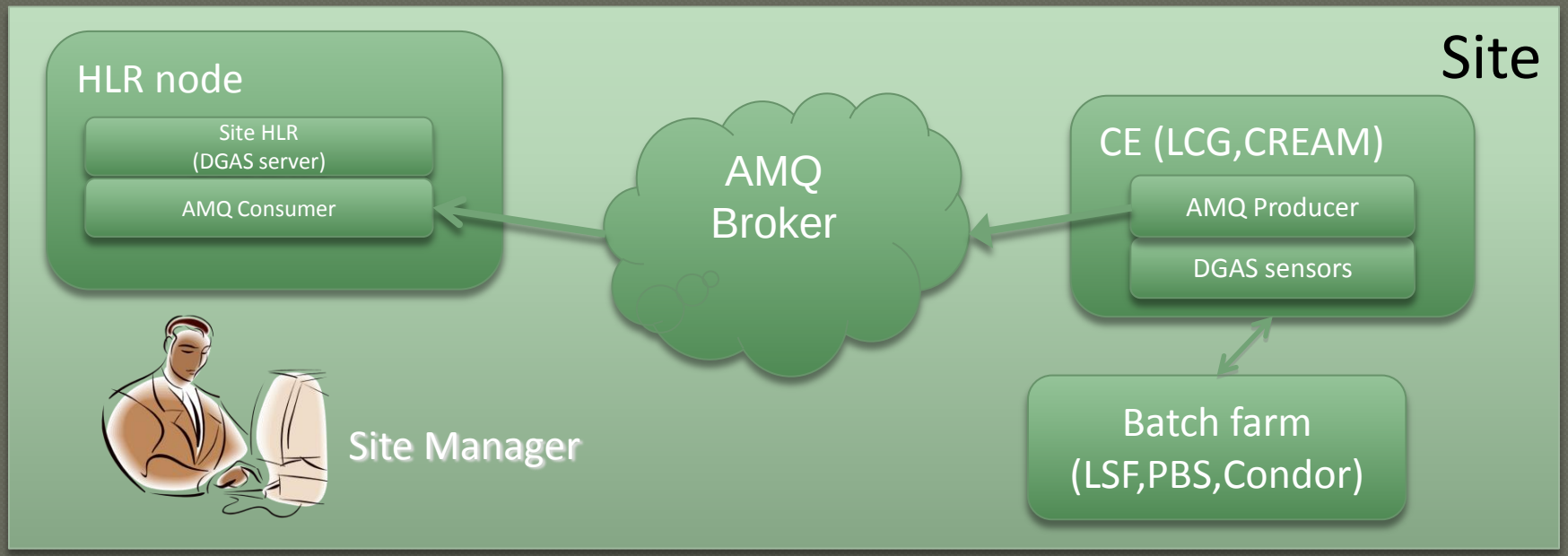
- Many improvements and new functionalities are needed in the medium to long term:
  - Use of sensors and servers with Active-MQ based messaging protocol.
  - Migration from the current (MYSQL based) communication protocol to activeMQ for DGAS2GOC tool.
  - Adoption of OGF-UR schema for Usage Records in addition to the legacy schema.
  - Make it possible to easily extend the contents of the Usage Rercord.
  - Use of different benchmark metrics in the Usage Record.
  - Fine grain normalization of CPU-related usage metrics.
  - Consolidate MPI job accounting.
  - Consolidate the Storage Accounting toolkit, introducing it in production.
  - Re-engineering of the Economic Accounting functionalities.

Legacy protocol



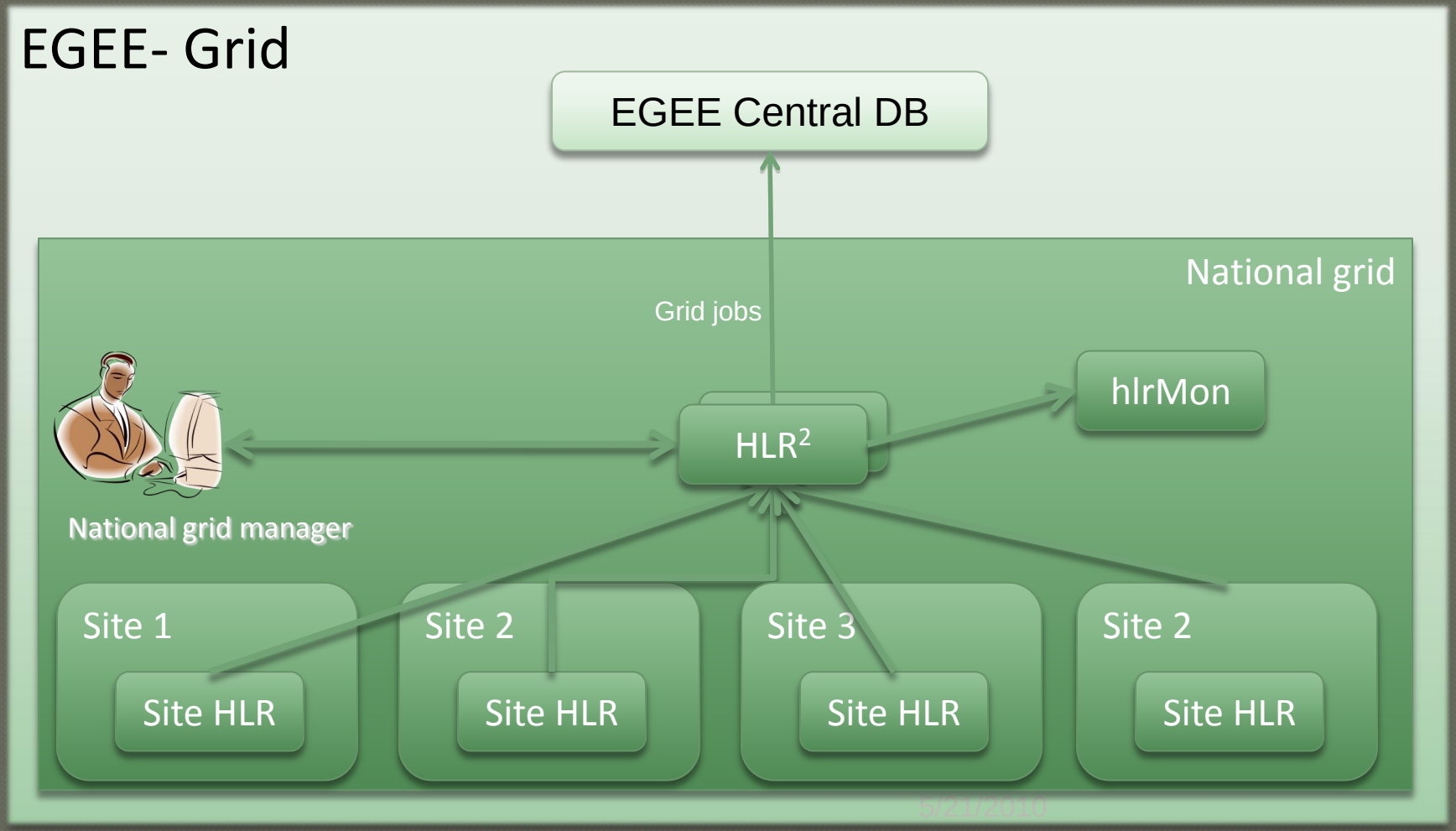
The basic, legacy, DGAS communication protocol is based upon a client-server infrastructure, with a double commit push protocol based on a GSS over TCP secured channel.

Active MQ

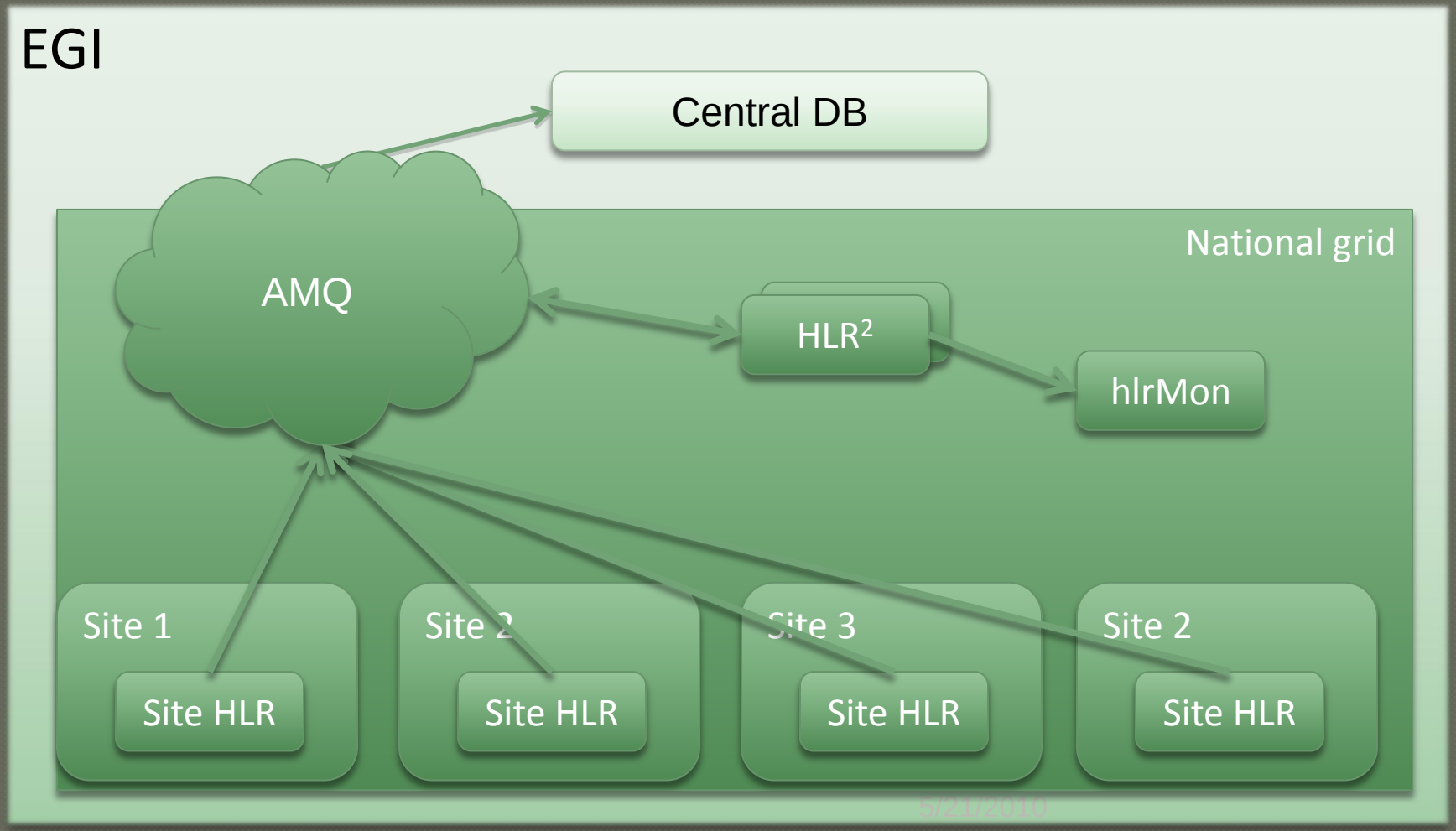


With the Active MQ model, the sensors act as message producers, while the HLR is a message consumer. In this scenario it is not possible to use an encrypted communication channel, so encryption is performed on a per-message basis.

Legacy communication protocol



Active MQ based commmunication



OGF UR producer

- One of the most important requirement to fulfill is interoperability.
- The adoption of the de-facto current standard for accounting message communication, active MQ, accomplishes just half of what we need.
- In the, hopefully, near future we will implement the AMQ-enabled DGAS, with a Usage Record Producer using the OGF Usage Record standard instead of the DGAS, legacy, UR.

Flexible UR schema

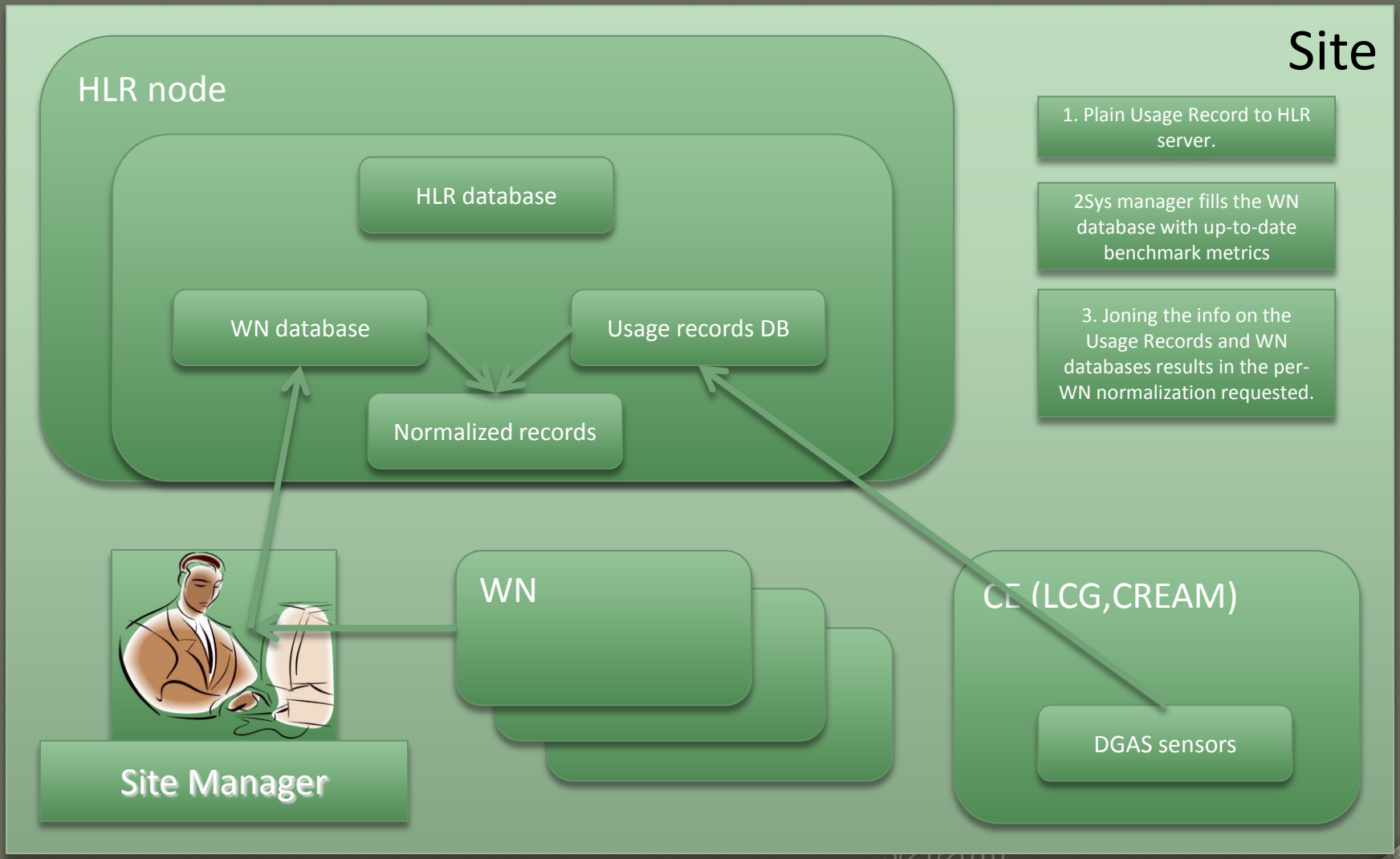
- Another important improvement will be the possibility for the system manager to modify the Usage Record schema and contents, without modifying the source code.
- This makes it possible to modify its contents (at least a part of), and add new contents.

WN-based Normalization

- Requirement: allow for fine grain (per WN) normalization of CPU accounting metrics on etherogeneous farms.
- Server side, from an HLR perspective, all that is needed is the availability of the WN hostname(s) information.
- A WN DB is being designed to address this requirement, see Peter Solagna talk.

WN-based normalization

WN database approach



Conclusion

http://www.to.infn.it/dgas