AGIS:
ATLAS Grid Information System

Alexey Anisenkov (BINP)

Information System TF meeting, 12 November 2015

Topics we try to cover

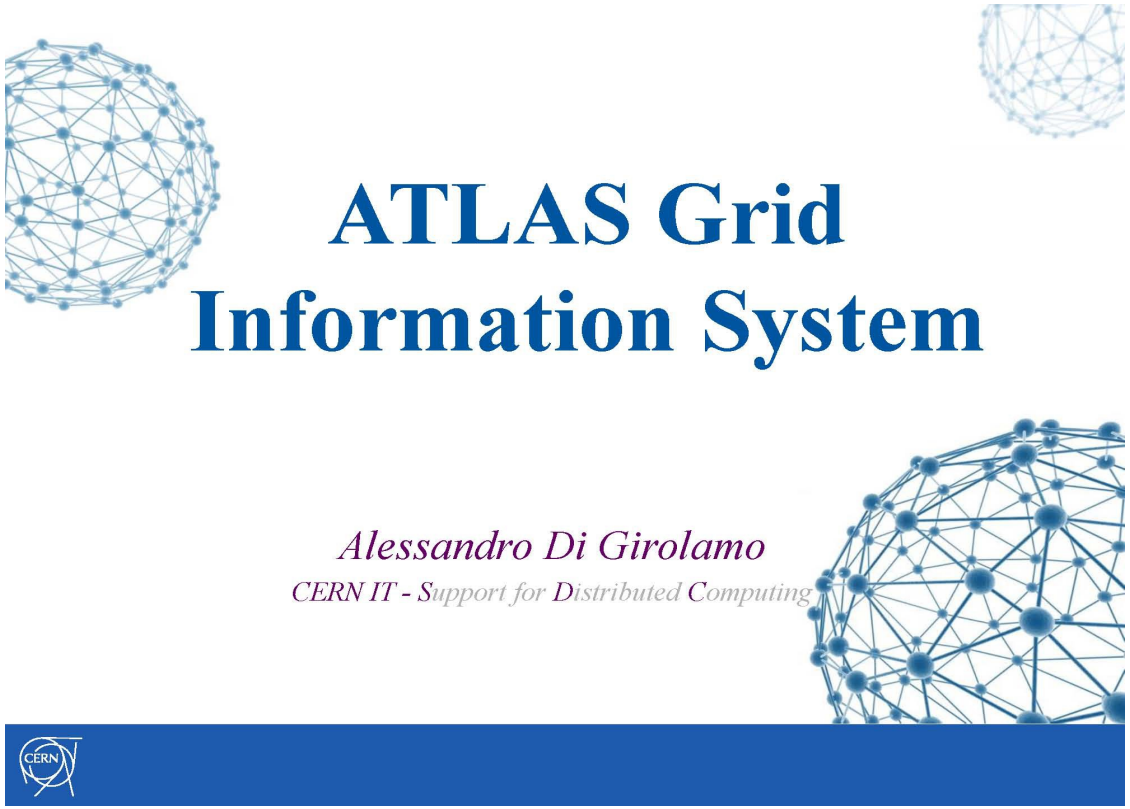
- General info about AGIS
  (highlights from Alessando's talk presented at NEC2015)

- Integration of new formats into AGIS:
  - GLUE2, HTCondorCE

- Recent AGIS developments
  - HPC/Cloud resources
  - object store
  - corepower metric
  - general system updates (ATLAS specifics isolation)

What is AGIS?
NEC' 2015 (Montenegro)

https://indico.cern.ch/event/408075/



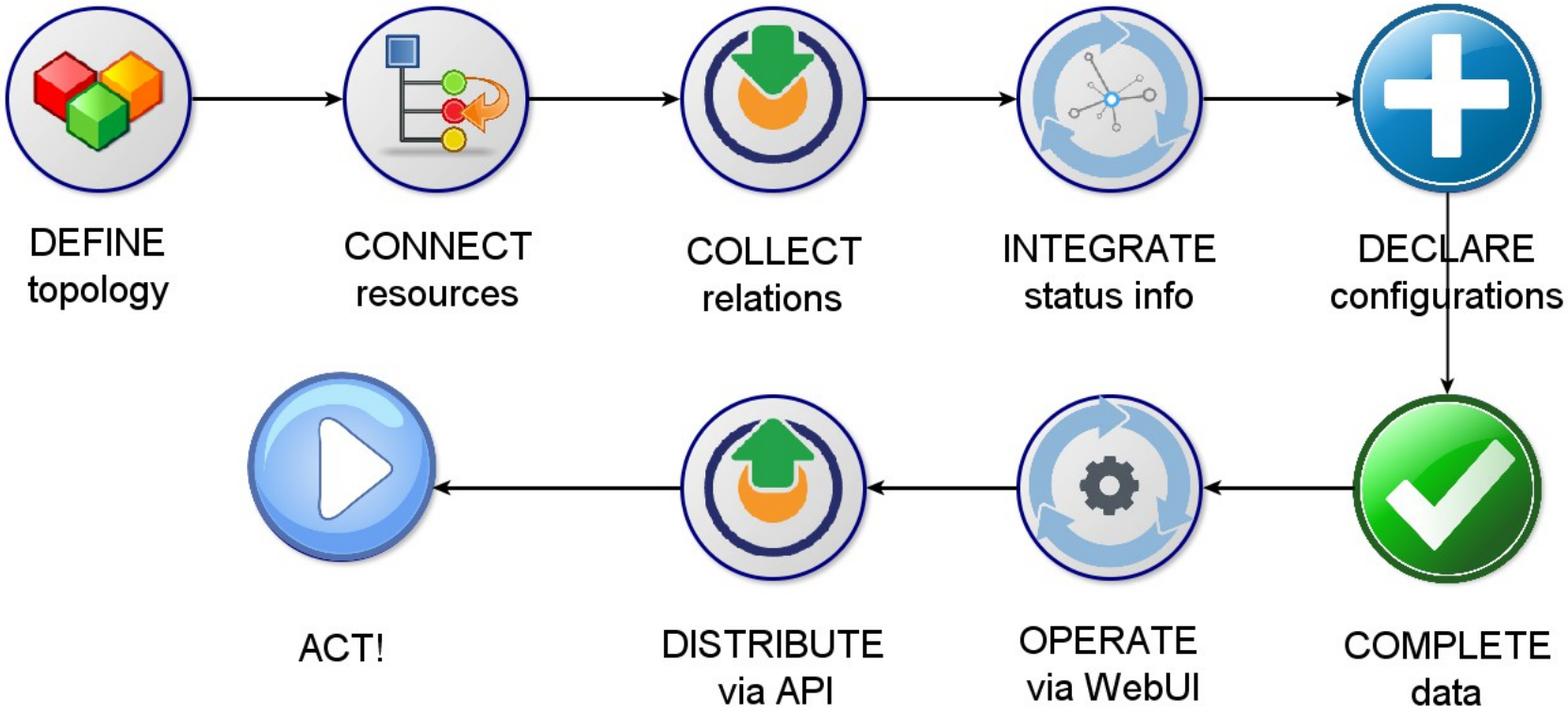
ATLAS Grid Information System

- Collect, integrate and allow to define Sites (Resource Centers), their services information, the resources configuration and status
  - Static information (lifetime of the service) and semi-static (e.g. downtimes).
- Describe and expose the ATLAS topology
- AGIS in few lines:
  - DB backend: persistent storage and changes history
  - webUI and collectors
  - REST API (POST and GET)

The AGIS project

- Started "many" years ago (>5)
  - Strongly wanted by ATLAS Computing management
  - Several groups involved
- Project leadership (stable since ~5 years)
  - Responsibility shared between CERN-IT/SDC and BINP (Novosibirsk)
  - Alexey Anisenkov as Technical Coord, myself as Project Leader
- Several persons contributed over the years
  - Dubna; summer students; experts from CERN-IT/SDC …

AGIS in few words:
Key capabilities of the system



AGIS collectors and webUI

- The list of sites supporting ATLAS is defined in AGIS
  - Procedures for new sites joining ATLAS:
    - Within WLCG for pledged resources
    - ATLAS International Computing Board for Tier3s (including unpledged HPC and CloudResources)
- For each site AGIS get the list of services from GOCDB and OIM
  - Mainly AGIS uses them as Service Registry for Service Discovery
- Service detailed information automatically collected or manually updated
  - E.g. CE from BDII, Storages described manually by SiteAdmins into AGIS
- webUI
  - Edit/Update specific service information
  - Add new services (if needed, e.g. HTTPS storages, ObjectStore…)
  - Define and “attach” new ATLAS specific objects

Service Discovery

- Today AGIS is both kind of
"service discovery" and the "detailed info" framework
  - "Standard" Grid sites: details fetched from OIM/GOCDB
  - Other resources like Cloud/HPC are directly defined in AGIS
  - Other services (ObjectStore) and specific configurations (FrontierSquid, FAX redirectors) are also directly defined in AGIS

- If a ***central*** Service Discovery tool would exist, it could be very useful
  - n.b. We do not suggest to have an ***aggregator*** of already existing tools, we suggest that we take everyone on board and we have just ***ONE single framework*** where SiteAdmins (SiteResponsible) can declare their sites/services/resources.

- Both static info and dynamic data declared in AGIS:
more in few slides from now

AGIS centric view

- AGIS is a fundamental piece of ATLAS Computing
  - Under the hood for the non-expert
    - But does not mean less important
  - "Fight" each day (monthly?) to avoid people/framework diverging
    - Continuous gathering of new requirements and development
- Key point: the ADC components "speak the same language" through AGIS:
  - Panda, Rucio, monitoring Dashboards, pilot, Frontier-Squids ...
    - You add e.g. one new storage you do not need to edit configs in N different frameworks, all within AGIS!
  - "Fast" complete integration of new technologies within ADC

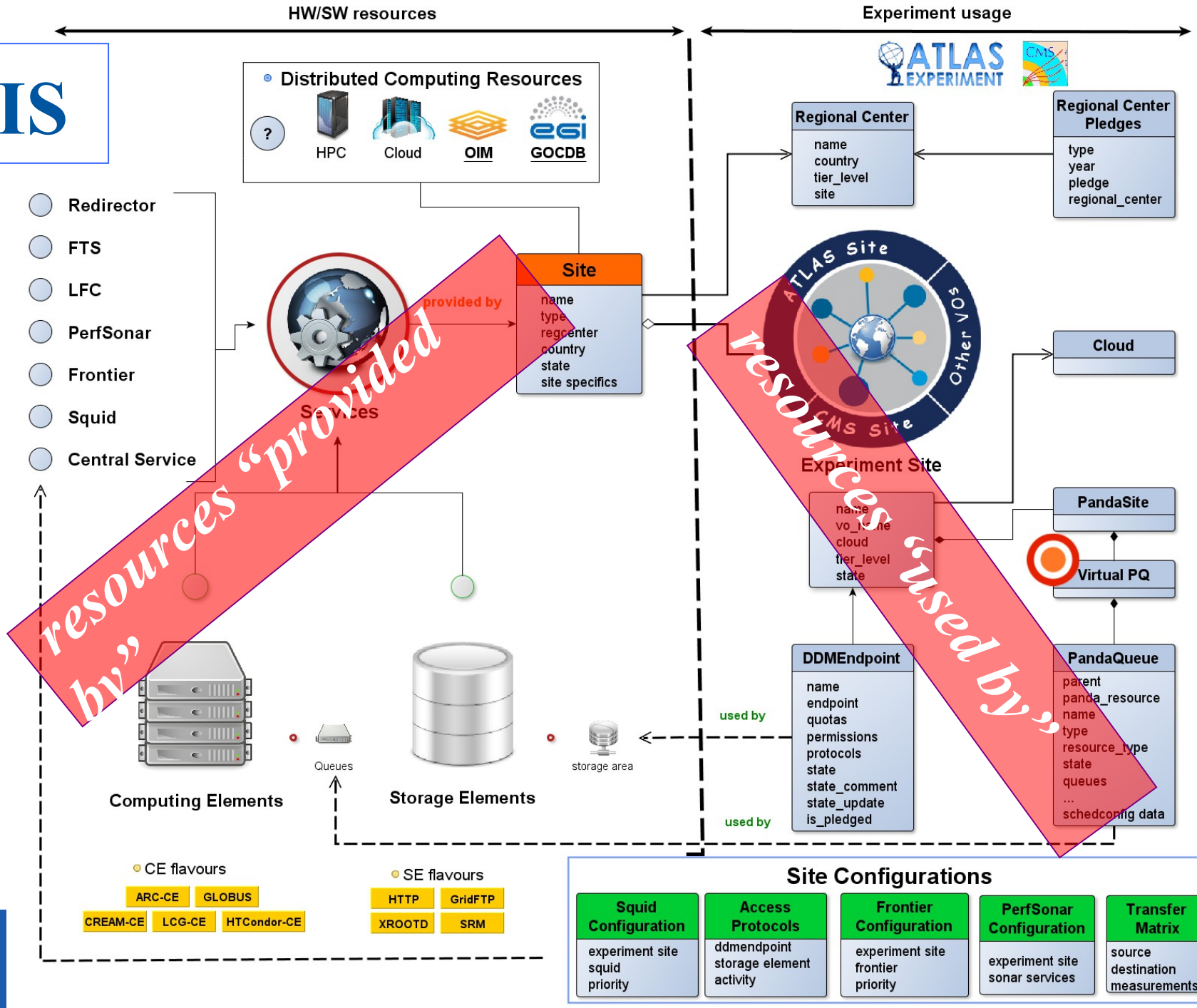
Fundamental concept of the system

- Clear distinction between resources
  provided by (Sites) and resources *used by* (Experiments)

- Establish relationship between resources to Experiment objects

More details: next slide

AGIS


HW/SW resources
Experiment usage
Distributed Computing Resources
HPC
Cloud
OIM
GOCDB
Redirector
FTS
LFC
PerfSonar
Frontier
Squid
Central Service
Services
provided by
Site
name
type
regcenter
country
state
site specifics
Regional Center
name
country
tier_level
site
Regional Center Pledges
type
year
pledge
regional_center
ATLAS Site
Other VOs
CMS Site
Experiment Site
name
vo_name
cloud
tier_level
state
Cloud
PandaSite
Virtual PQ
PandaQueue
parent
panda_resource
name
type
resource_type
state
queues
...
schedconfig data
DDMEndpoint
name
endpoint
quotas
permissions
protocols
state
state_comment
state_update
is_pledged
used by
used by
Queues
storage area
Computing Elements
Storage Elements
CE flavours
ARC-CE
GLOBUS
CREAM-CE
LCG-CE
HTCondor-CE
SE flavours
HTTP
GridFTP
XROOTD
SRM
resources "provided by"
resources "used by"
Site Configurations
Squid Configuration
experiment site
squid
priority
Access Protocols
ddmendpoint
storage element
activity
Frontier Configuration
experiment site
frontier
priority
PerfSonar Configuration
experiment site
sonar services
Transfer Matrix
source
destination
measurements

AGIS

- Collect integrate and allow the sites services information, resources configuration and status. E.g.:
  - CE host, type, jobmanager, port, queues (with details)
  - Storage element host, type, protocols, path, etc
  - PandaQueue (aka PandaResource) config
  - DDMEndpoint (aka RSE Rucio Storage Element) configuration

- New needs (e.g. ObjectStore storage, HTCondorCE)?
  - Doable but not easy
  - Several interactions are needed to understand and clarify the requirements
  - Experience tells that we are able to get things into AGIS in timely manner (weeks to get them in production)
  - It is not just a click, no free lunch!

Feeding: Automatic & Manual

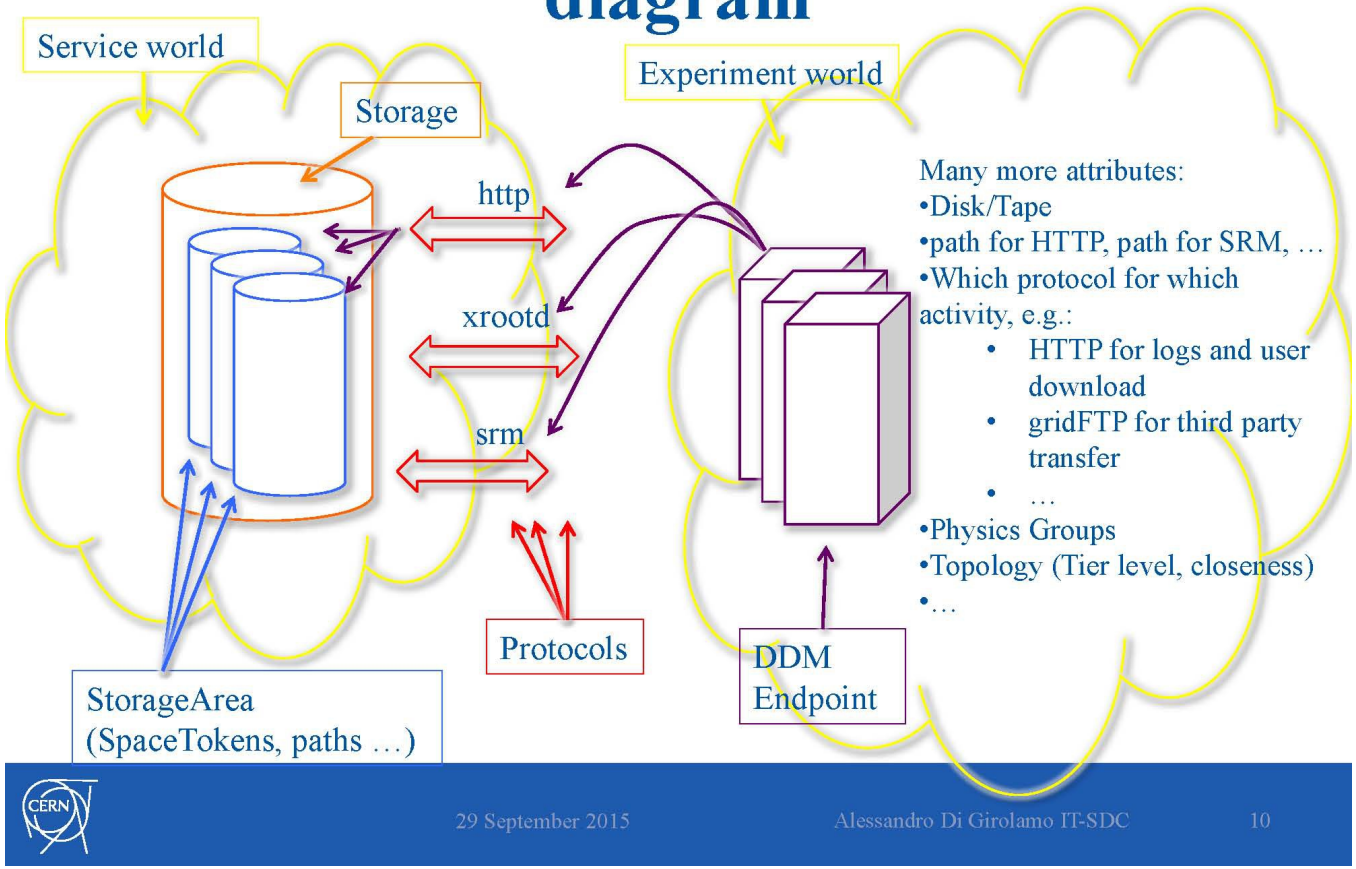
- Automatic as much as possible!
  - But not always possible
- Manual but with deep validation at insertion level!
  - Clearly defining the possibilities for SiteAdmins/Experts
    - Too much freedom sometimes is no good
  - How many new storage do we have per year? Very few!
    - Each change in a storage requires many changes in other frameworks
    - not something worth to automate
- REST API for both AGIS webUI and for users
  - Possibility to automate
- Common "Language"
  - AGIS is able to understand and speak Glue1.3 and 2, and also able to digest other kind of information
    - Agreeing on a common language (at least meaning of keywords) is important

Other formats or new resources? Yes.

- AGIS can be easily extended and used to describe new types of services and resources, recent examples:
  - New type of SE: Object Store resources required by ATLAS
  - Corepower metric (average core power per site), initially collected from the REBUS, propagated into AGIS and completed for missing T3 sites.

- OR/AND collects data from various sources, aggregates data into the information model if need:
  - HTCondor-CE implementation (services and queues declaration), data getting from restful OSG source.
  - New Glue2 collector: to be developed

Distributed Data Management Endpoint diagram



- Even if resource topology could be complex:
  - Operations and service declaration by SiteAdmins should not be complicated

SE/CE resources declaration (in BDII)

general thoughts based on AGIS experience

- *SE declaration*: can we make sure that **all** our Storages are properly described?

Our experience: many different declaration in BDII, most of them not wrong, it's just that there is a lot of flexibility → too much freedom, not enough clear instructions for “standard” sites

- This means experiments (AGIS) need to customize each query for each site!
  - we created a webUI and we solved the problem!

SE/CE resources declaration (in BDII)
general thoughts based on AGIS experience

- In general: the language is OK,
  but we need to have clear examples and procedure for site admins,
  we need to agree on the meaning of each word, etc

OR
may be simplify?

and restrict a lot of freedom ...

SE/CE resources declaration

a proposal to simplify publication by sites

sometimes things look overcomplicated due to high flexibility

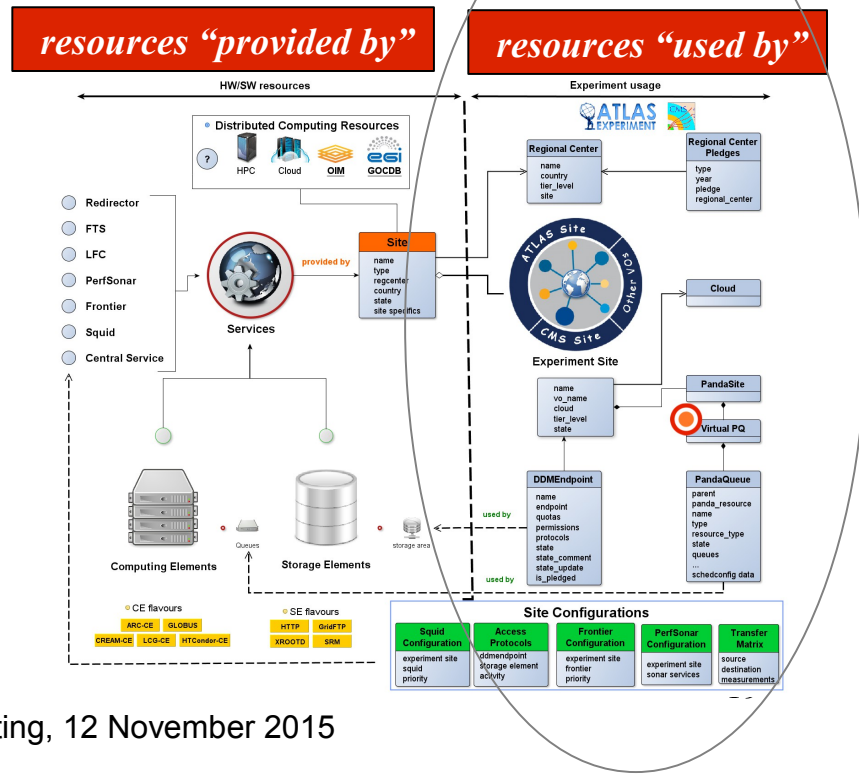
- An experiment (ATLAS as an example) really needs very small subset of attributes (currently available in BDII) to properly declare service or resource (~ 10 parameters)
  - e.g. CE topology definition: CE (*name*, endpoint, type, flavour, version, *status*, description, jobmanager) + localqueue details (name, max_cputime, max_wallclocktime, *status*)
- We propose to ask sites to publish/export (separately) only these actually used resource's attributes:
  - somewhere, e.g. providing any REST-full way (JSON, XML, other structured format) using httpd service
  - In predefined simplified format
  - GOCDB/OIM can store URL to such site's export
- So that experiments can fetch information they really need and sites simplify operations.

AGIS for other collaboration? Yes!

- Experiment view of resources ("provided by vs used by" concept)

- Doable to create experiment specific object ("used by" part) and implement Experiment specific configurations

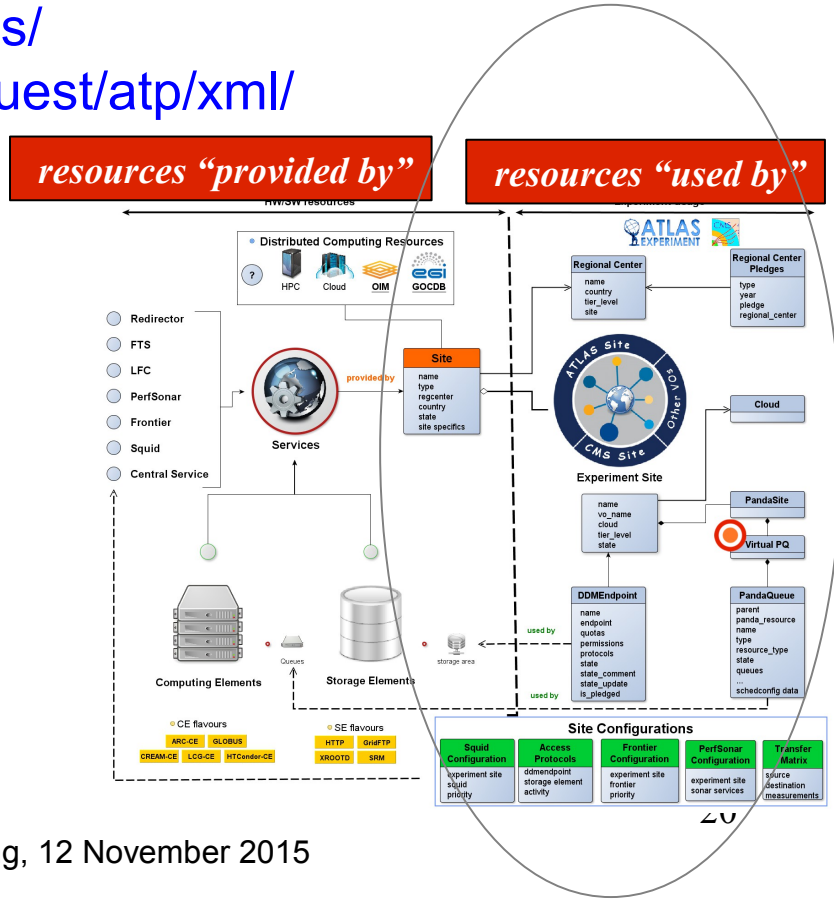
- All the rest can be "***practically***" get for FREE:
  - already implemented WebUI views
  - Customized form validations
  - Changes log
  - REST style GET/POST API
  - Others shared functionality provided by AGIS as a framework



we just have to agree and consider AGIS as a possible framework

AGIS for other collaboration? Yes!

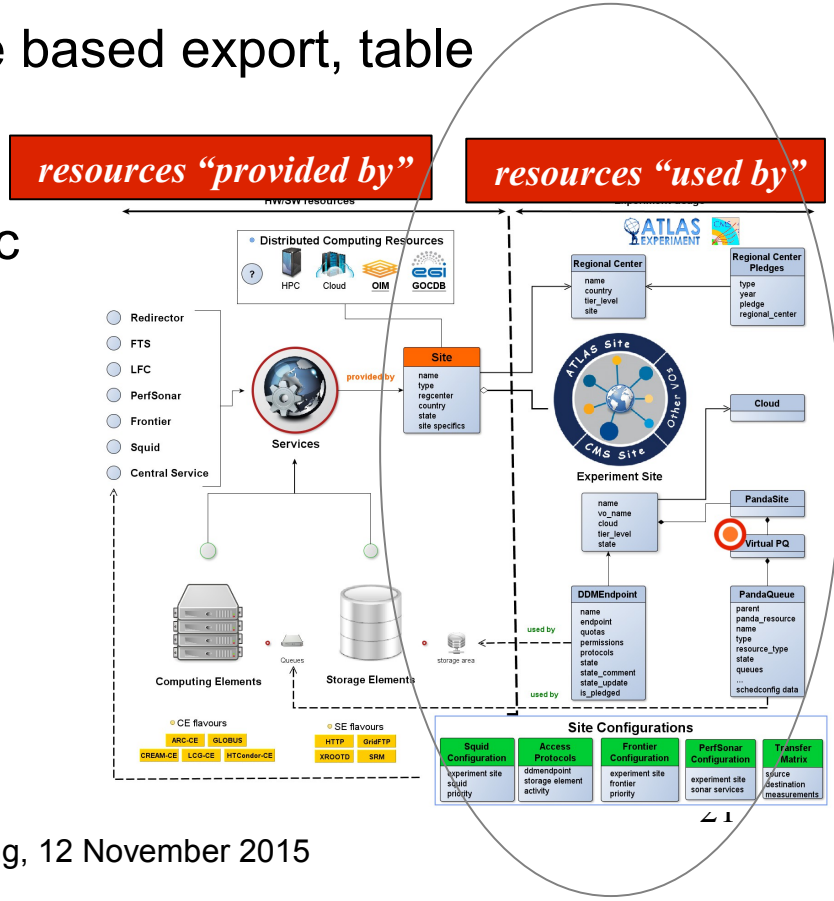
- We already tried AGIS as test IS for CMS to prove the concept:
  - Standalone testbed CMS specific AGIS instance (WebUI/API)
    - No one word about ATLAS
    - http://cms-info-system.cern.ch/agis/
    - http://cms-api-system.cern.ch/request/atp/xml/
  - Covered use-cases:
    - declaration of CMS site topology
    - GlideinResources definition
    - site pledges.



we just have to agree and consider AGIS as a possible framework

AGIS for other collaboration? Yes!

- We are evolving AGIS core and making all ATLAS specifics as a plugin(s) for the system:
  - Customization of WebUI
  - Unification of basic WebUI views (tree based export, table views, interactive widgets, etc)
  - HTML templates isolation
  - Completely isolation of all ATLAS logic
  - Special VO-specific build of sources.



Summary

- AGIS describe the topology and services of ATLAS Distributed Computing
  - Concept: “Provided by” vs “Used by”
  - Flexible: new objects/collectors can be added without major re-organization of the framework
    - But not enough to snap your fingers, need to be properly architected
  - Clear goals, technical expertise to make it happen, support from the management are some of the key aspects which made AGIS useful and used
- Storages and Federations:
  - Need common strategy and solid agreements