Synopsis Project
A « cloud ready » architecture

Fabien MARTY, june 2011

{SYNOPSIS}

METEO FRANCE
Toujours un temps d'avance

Plan

- If you missed something...
- Technical and functional (new) specifications
- Technical foundations
- Diving into the server architecture
- Hardware infrastructure

Plan

- **If you missed something...**
- Technical and functional (new) specifications
- Technical foundations
- Diving into the server architecture
- Hardware infrastructure

If you missed something...

- After a 18 months pilot project about SOA and OGC applied to operationnal forecasting tools...
- ... Météo-France and MFI launched the Synopsis project mid 2010 to get :
  - A single workstation for “advanced forecasting” (with a progressive and smooth transition from Synergie)
  - A “light” solution (web-based) for other needs
  - Sharing the same business OGC server components

{SYNOPSIS}

mfi
METEO FRANCE INTERNATIONAL

If you missed something...

- Now :
  - We are ending complete specifications and ergonomic studies
  - We are releasing a first operational version for a specific need

If you missed something...

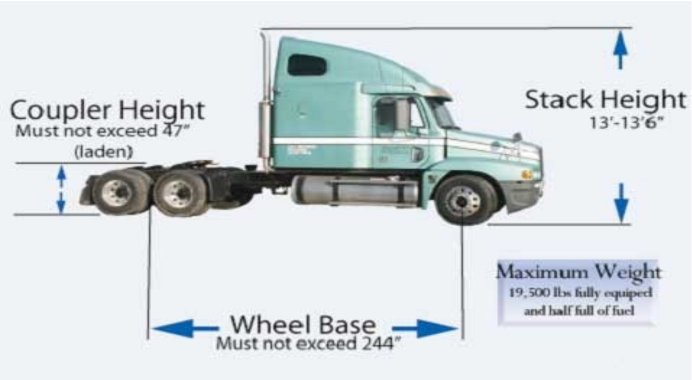
- In september :
  - Operational client for some of our forecasting objects
  - Operational OGC services for radar, satellite and cartography
- Roadmap :
  - Mid 2013 : a complete product for most visualization needs
  - Mid 2015 : everything is done

Plan

- *If you missed something...*
- **Technical and functional (new) specifications**
- Technical foundations
- Diving into the server architecture
- Hardware infrastructure

Technical (new) specifications

- OGC and SOA
- multi-platform and auto-deploying client
- Flexible architecture which must work :
  - As standalone behind a satellite receiver
  - As an high performances « cloud » service
    - horizontal scalability
    - no service interruption upgrades
  - As an hybrid thing with several levels of data access



METEO FRANCE
Toujours un temps d'avance

Functional (new) specifications

- Interoperability in both directions
- Zooming and panning without any constraints
- Adaptive GUI depending on :
  - Forecaster profile
  - Really available datas on the server side for the current context
- Customizable GUI at the user level

Plan

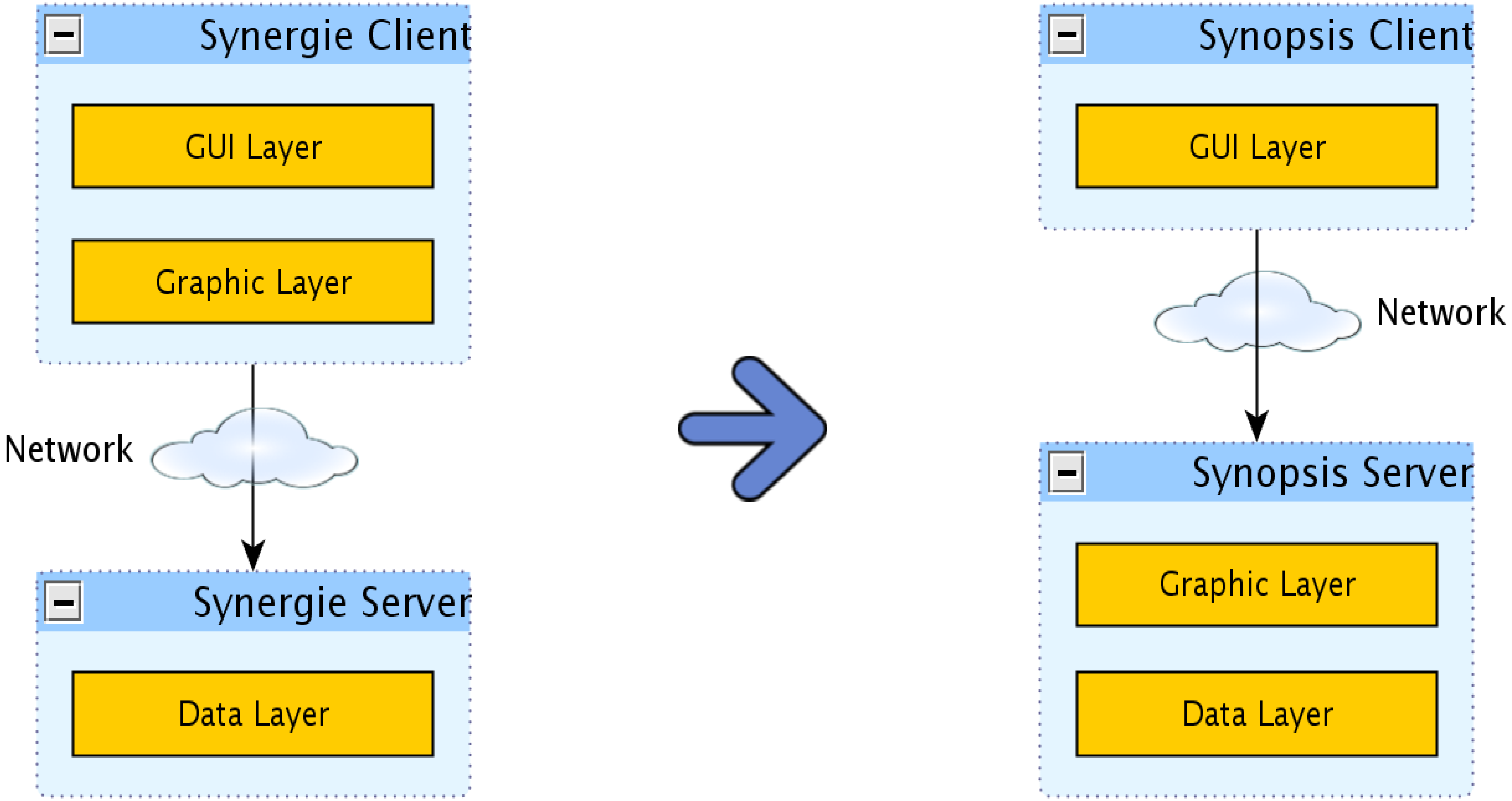
- *If you missed something...*
- *Technical and functional (new) specifications*
- **Technical foundations**
- Diving into the server architecture
- Hardware infrastructure

Technical foundations

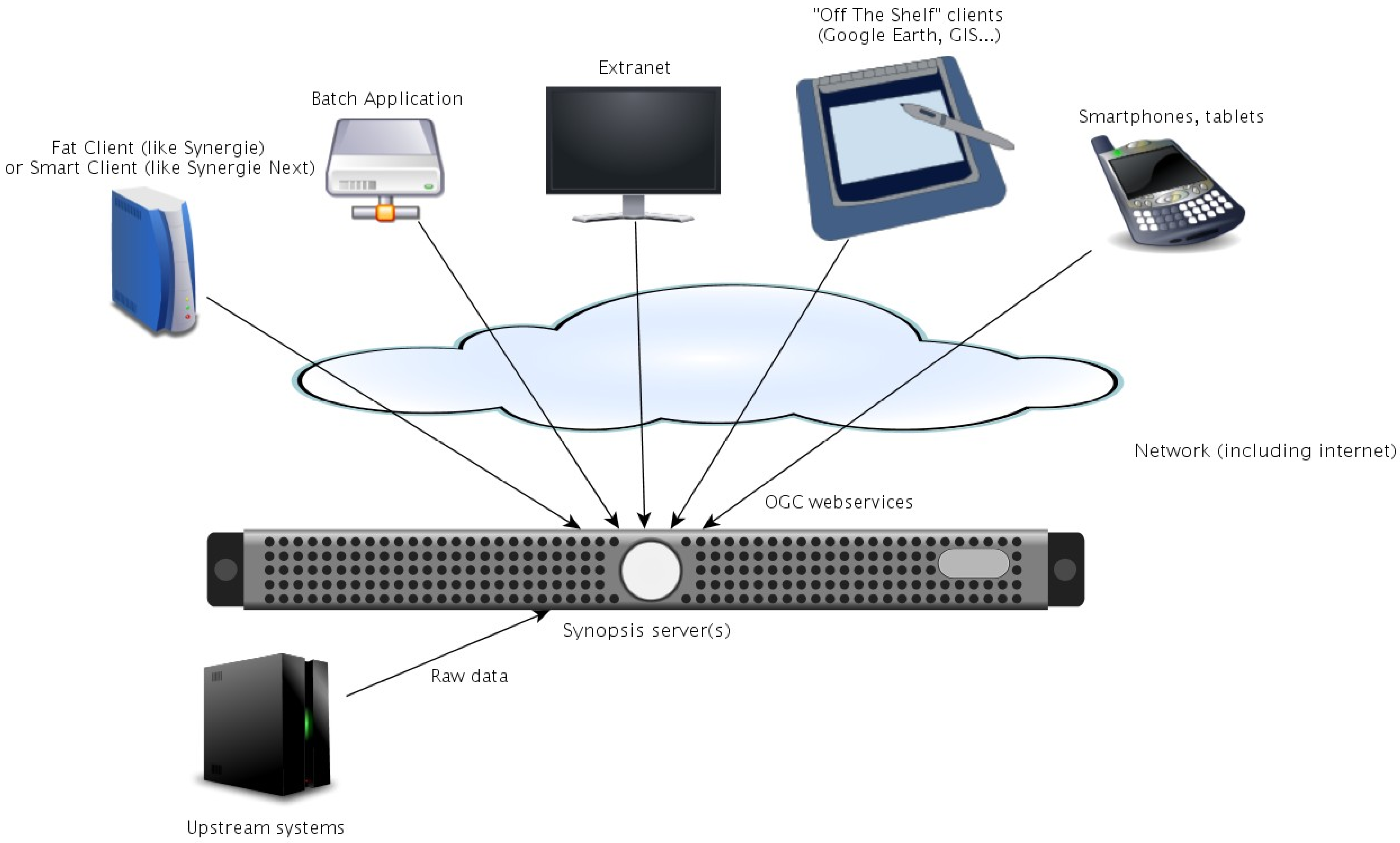
- (1) A center of gravity on the server side
- (2) « Java Web Start » thin client based on :
  - WMS layers concept
  - Netbeans Platform (RCP)
- (3) « Linux only » server architecture based on :
  - Opensource components
  - Web Oriented Architecture (WOA) : SOA reshaped and simplified by RESTful concepts

Technical foundations

- (1) A center of gravity on the server side

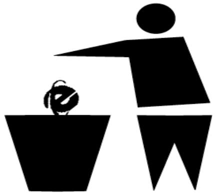


Technical foundations



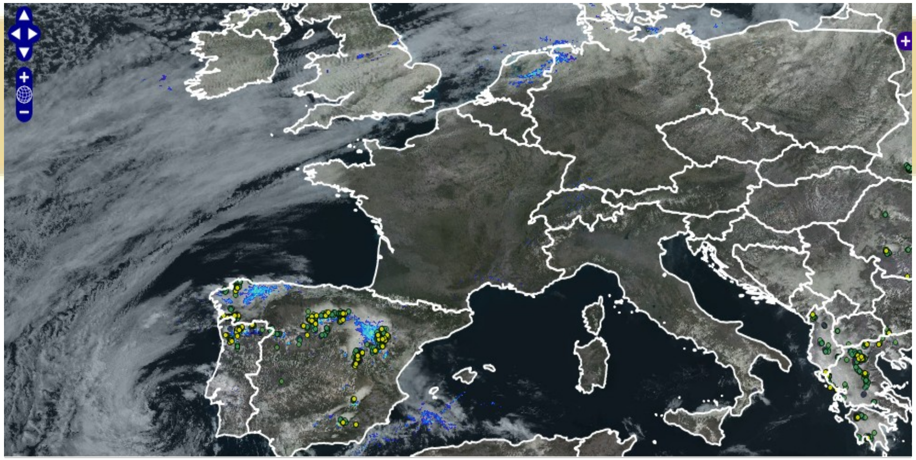
Technical foundations

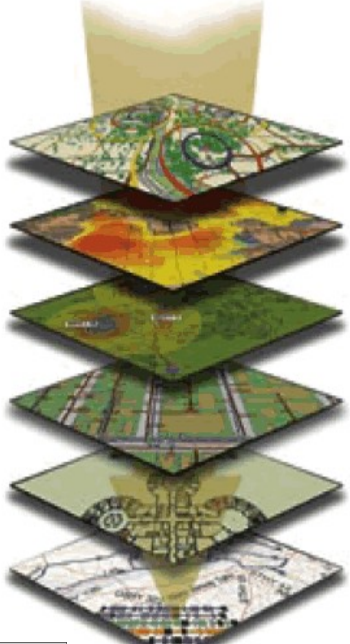
- (2) « Java Web Start » thin client :
  - (really) Multi-Platform
  - Automatic deployment
  - No fight with « Internet Explorer » !
  - Generic workstation :
    - No data, profile or configuration stored locally
    - Everything comes from the server infrastructure depending on the user login/profile
    - Only cache
  - **The server infrastructure acts as a « cloud service »**

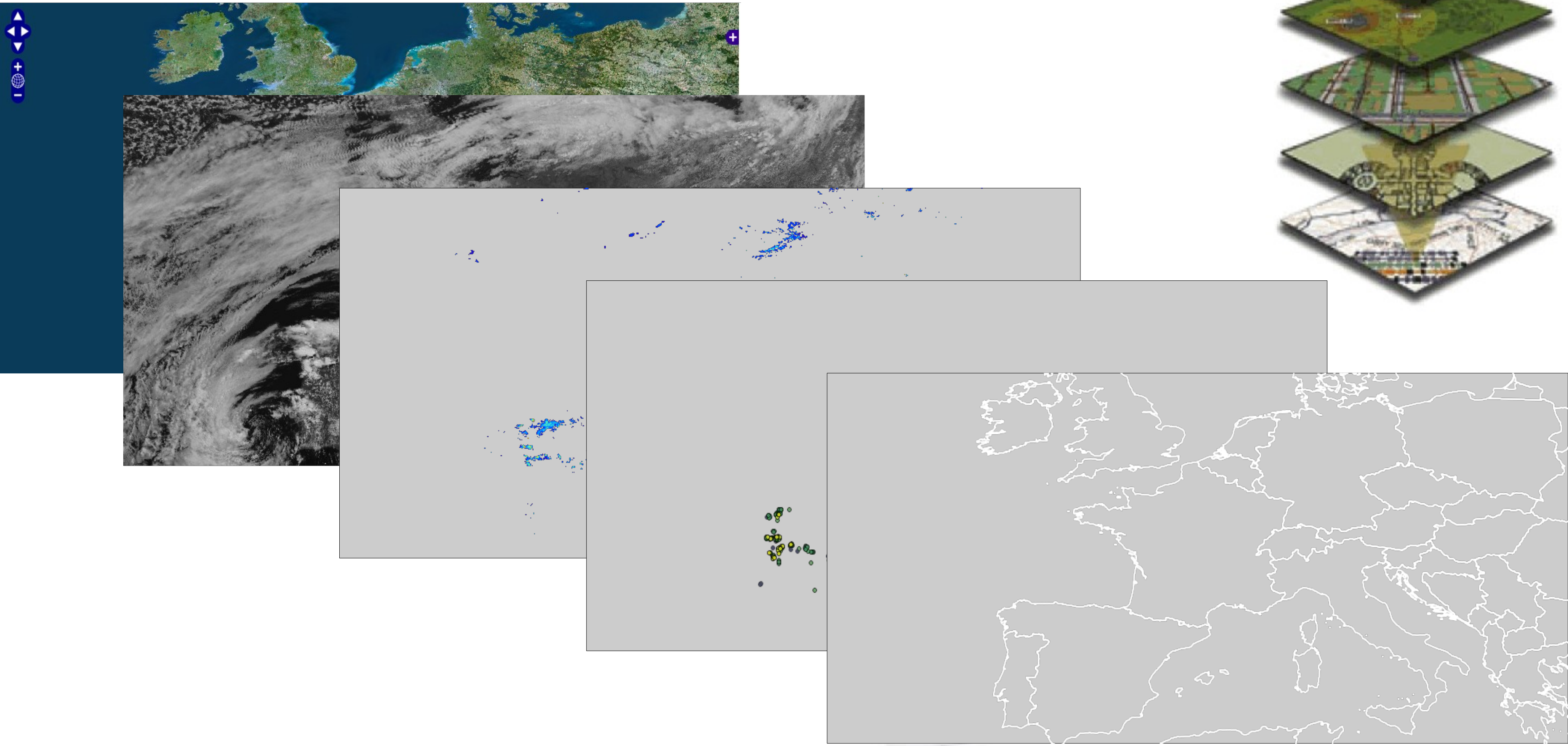


Technical foundations

- (2) « Java Web Start » thin client :
  - It requests plots to the server infrastructure as unit WMS layers...

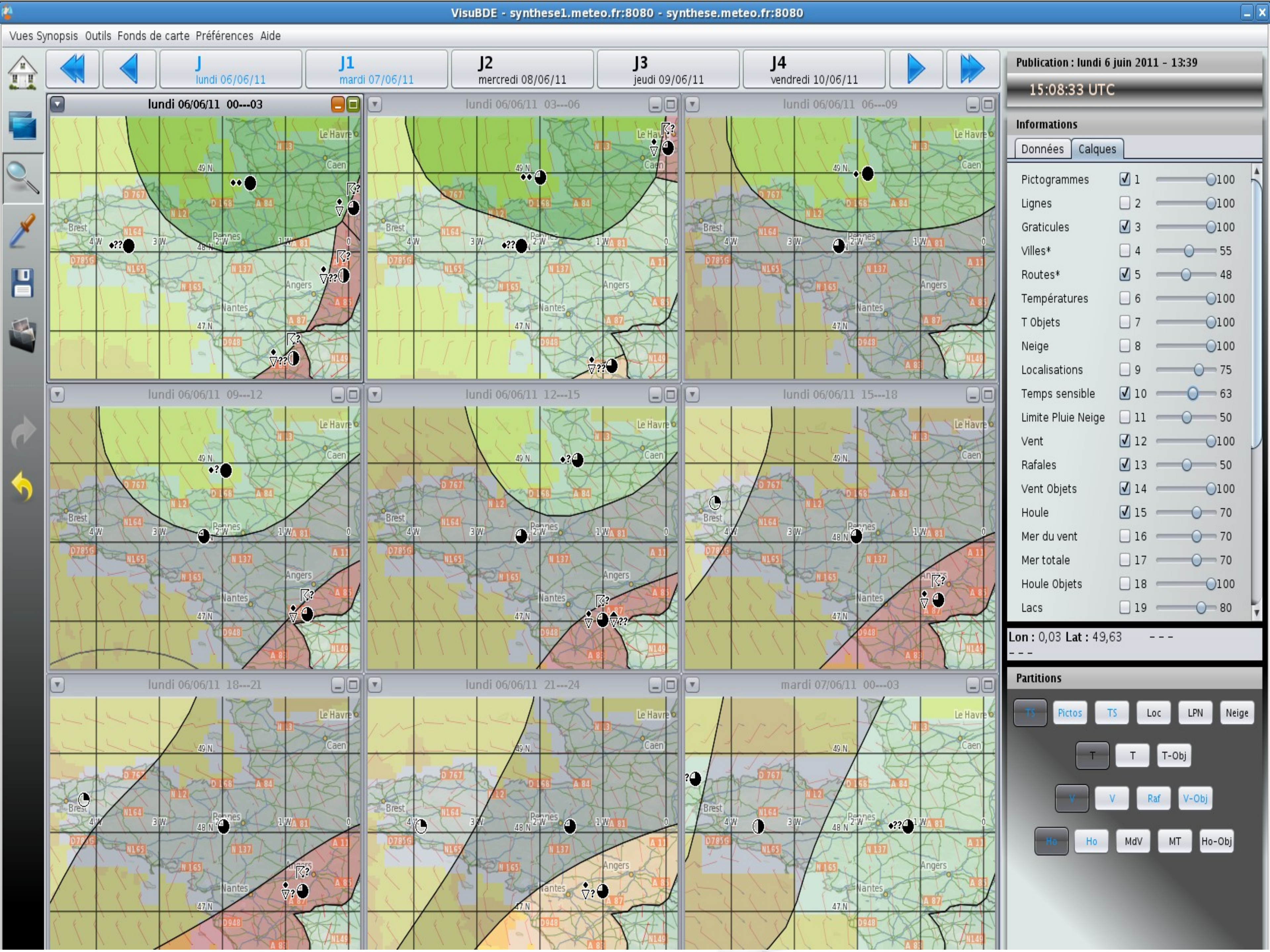






Technical foundations

- Consequences : a lot of WMS requests for a single user playing with something like this...



Publication : lundi 6 juin 2011 - 13:39

15:08:33 UTC

Pictogrammes	☑	1	100
Lignes	□	2	100
Graticules	☑	3	100
Villes*	□	4	55
Routes*	☑	5	48
Températures	□	6	100
T Objets	□	7	100
Neige	□	8	100
Localisations	□	9	75
Temps sensible	☑	10	63
Limite Pluie Neige	□	11	50
Vent	☑	12	100
Rafales	☑	13	50
Vent Objets	☑	14	100
Houle	☑	15	70
Mer du vent	□	16	70
Mer totale	□	17	70
Houle Objets	□	18	100
Lacs	□	19	80

Lon : 0,03 Lat : 49,63 - - -
- - -

Technical foundations

- (3) « Linux only » server architecture based on opensource components
  - **Languages** : Python, Java, C
  - **Additional libraries** : Gnome and Apache
  - **Database** : PostgreSQL / PostGIS
  - **Bus, Cache** : Redis
  - **Web** : nginx, gunicorn, django
  - **Specific tools** : Mapserver, GDAL, Magics++, GribAPI, Proj4

Plan

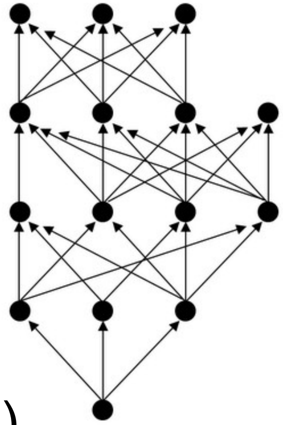
- *If you missed something...*
- *Technical and functional (new) specifications*
- *Technical foundations*
- **Diving into the server architecture**
- Hardware infrastructure

into the server architecture

- Choices :
  - Web Oriented Architecture (WOA)
    - SOA reshaped and simplified by RESTful concepts
  - The whole infrastructure is divided in 10 modules
  - Each module is :
    - A dedicated unix user
    - A dedicated home directory
    - A dedicated installation package (RPM)
  - Modules communicate each-other only throw the network (enforced by very restrictive unix permissions)
  - => So you can deploy most modules on different servers without any complications

Diving into the server architecture

- 10 modules :
  - 3 « dependencies modules » which must be installed on every server of the cluster
    - **synext** : SYNopsis EXTernal
      - External free libraries and binaries
    - **syndev** : SYNopsis DEVelopment
      - Libraries and binaries for development only
    - **syncom** : SYNopsis COMmon
      - Common libraries and binaries (maintained by us)
    - **They don't run anything !**
    - We would like to release them as free software

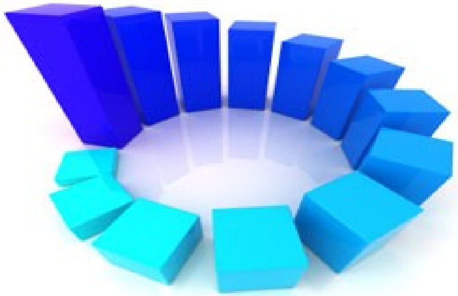


Diving into the server architecture

- 10 modules :
  - 3 modules not « horizontal scalable » :
    - **synbase** : SYNopsis dataBASE
    - **synadmin** : SYNopsis ADMin
    - Just classic master/master (by choice)
    - We plan to use sharding if the base becomes the bottleneck
    - **syndata** : SYNopsis DATA
      - Pre-processing of incoming raw datas
      - One instance per synbase module

Diving into the server architecture

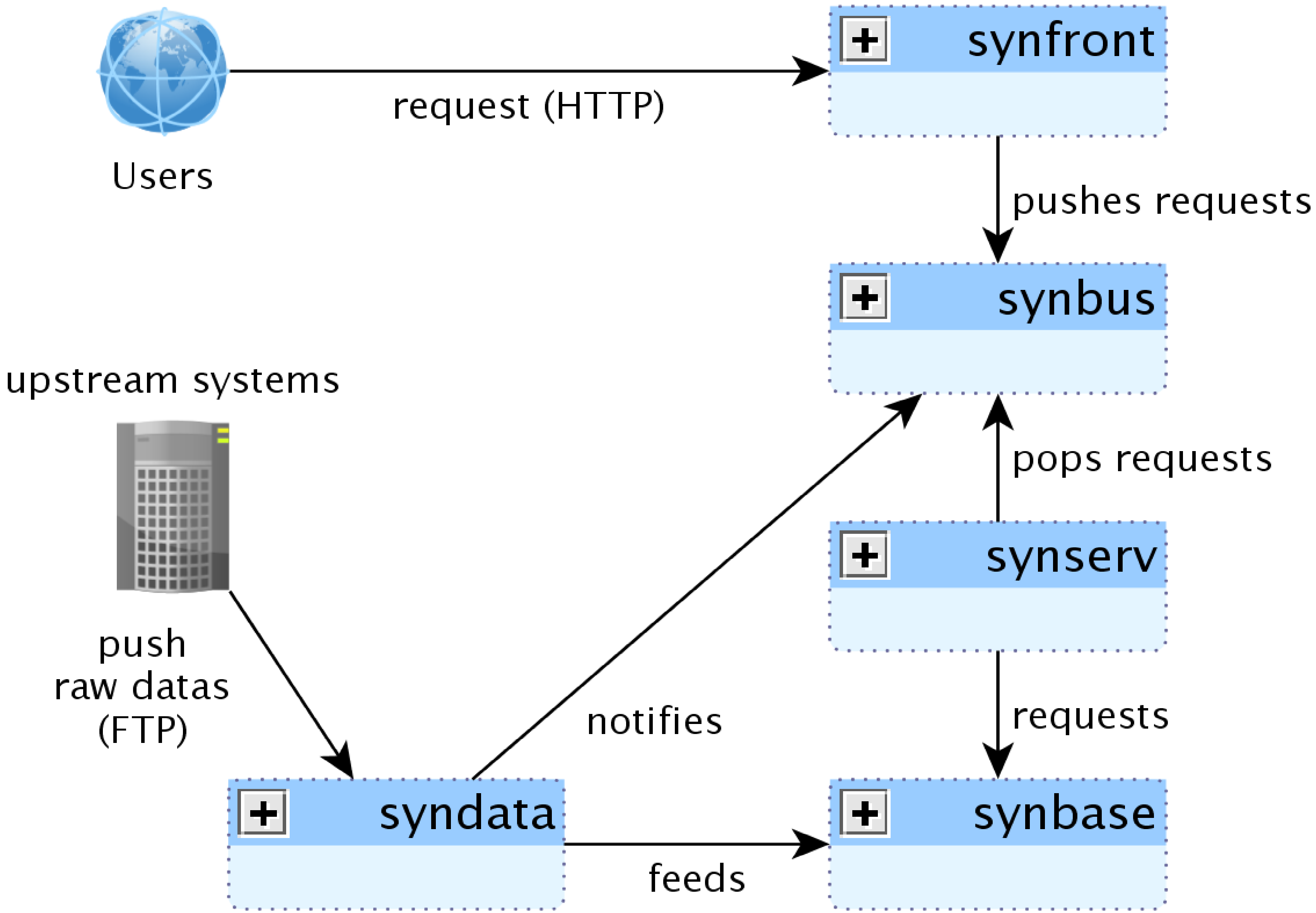
- 10 modules :
  - 3 modules « horizontal scalable » :
    - **synfront** : SYNopsis FRONTend
      - Each incoming request passes through this module
      - Output cache, security checks, routing
    - **synbus** : SYNopsis BUS
      - Communication bus between modules
      - Loosely coupling
    - **synclient** : SYNopsis CLIENT
      - Little web portal, distributes the Java Client



METEO FRANCE
Toujours un temps d'avance

Diving into the server architecture

- 10 modules :
  - 1 module « hot horizontal scalable » :
    - **synserv** : SYNopsis SERVices
      - The main module
      - Deals with non trivial requests
      - **Draws requested maps !**
      - We can hot add or remove instances of this module **with no extra-configuration**



Very simplified view of the Synopsis server architecture

Diving into the server architecture

- Technical implementation
  - synfront and synbus modules
    - made to deal with more than 10000 concurrent connections by instance (« C10K problem »)
    - Mix of :
      - « off the shelf » daemons (nginx, redis)
      - Custom glue code, made in C with only « event driven » network programming
    - Custom COMET daemon to notify clients from the server through an HTTP connection (opened by the client)

Diving into the server architecture

- Technical implementation
  - synbase module
    - PostgreSQL/PostGIS with daily dynamic table partitionning
    - WEBDAV repository for « medium and big files »
  - syndata module
    - Linux inotify system (to avoid polling)
    - 5 stages pre-processing
    - « extended attributes » on files to keep the traceability of the pre-processing

Diving into the server architecture

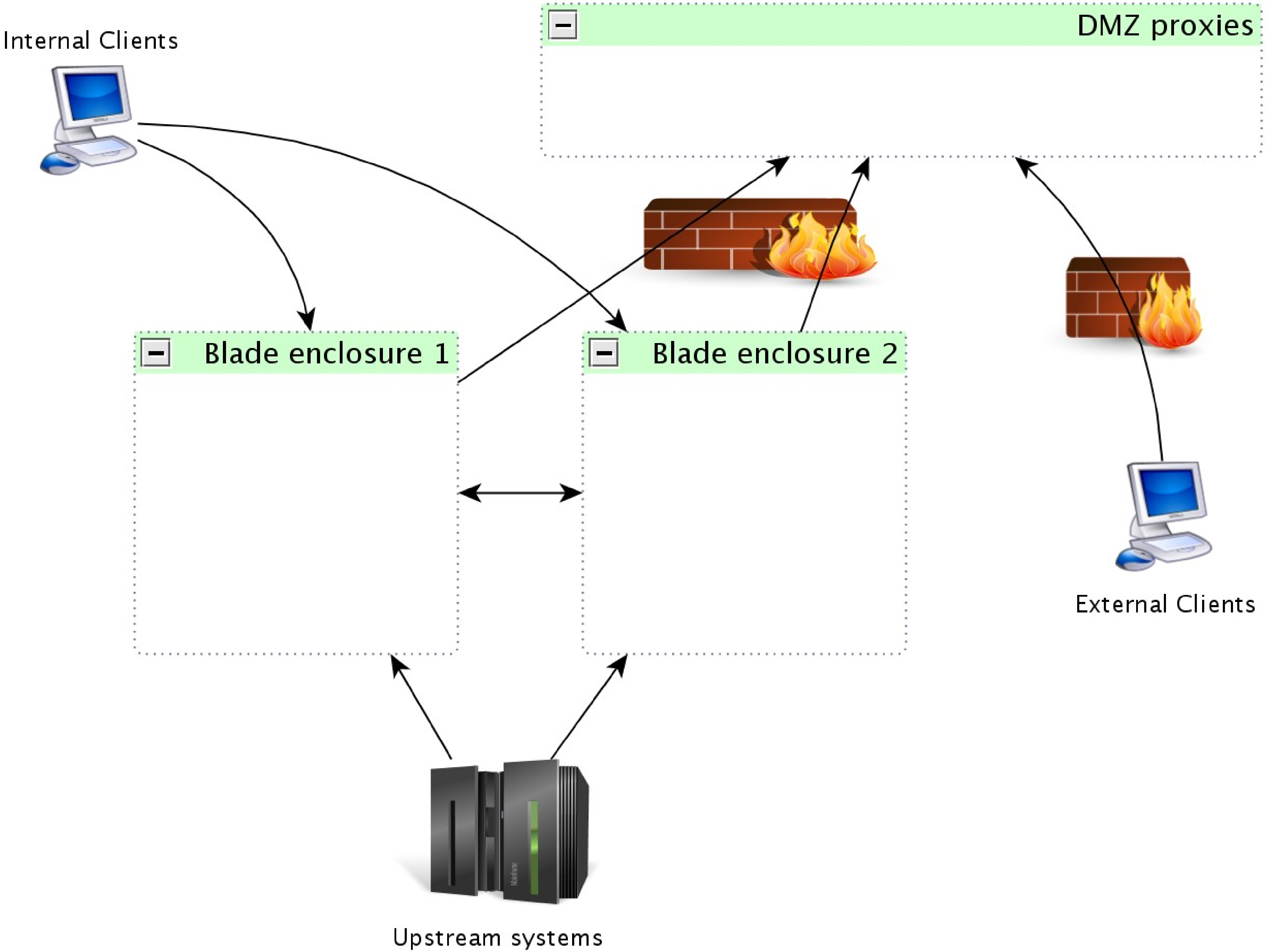
- Technical implementation
  - synserv module
    - The number of concurrent connections on a given module is limited by design and configuration
    - Services calls are just RESTful HTTP requests processed by :
      - A WSGI Python/Django stack
        - Mapserver, GribAPI and Magics++ are used as python libraries
      - (soon) or a JAVA/Tomcat stack
    - A distributed « in memory » application cache (with a sophisticated automatic invalidation at data arrival)
    - Custom filesystem (FUSE) :
      - To have a transparent access for the (potentially remote) webdav repository
      - To have a transparent cache

Plan

- *If you missed something...*
- *Technical and functional (new) specifications*
- *Technical foundations*
- *Diving into the server architecture*
- **Hardware infrastructure**

Hardware infrastructure

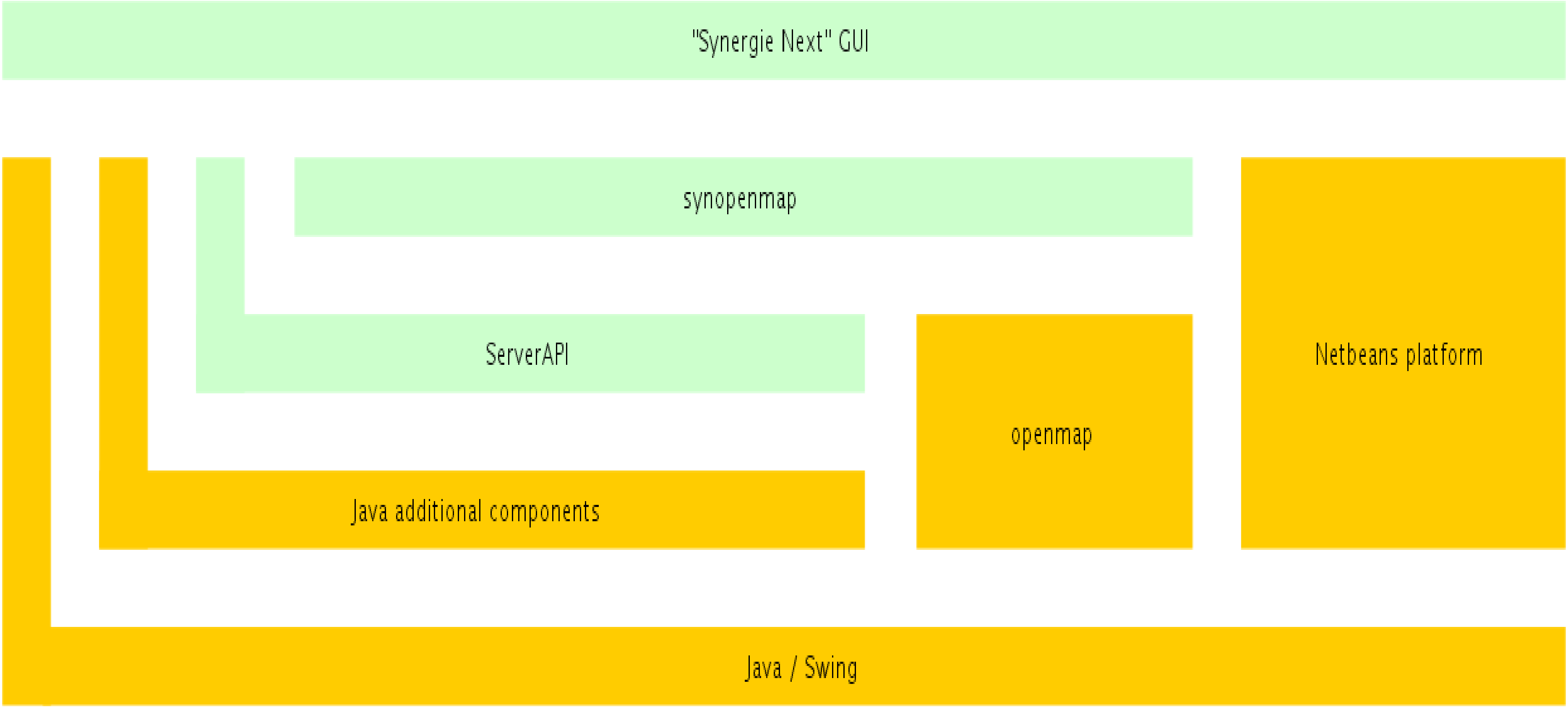
- Main ideas :
  - As flexible as the software architecture
  - High availability
  - No duplication for DMZ
- Choices (for starting up the service) :
  - 2 half full blade servers enclosure
  - Just 2 little proxies in DMZ (frontend and bus)
  - Each drawing blades can be dedicated for DMZ, for LAN or shared


DMZ proxies
Internal Clients
Blade enclosure 1
Blade enclosure 2
External Clients
Upstream systems

Plan

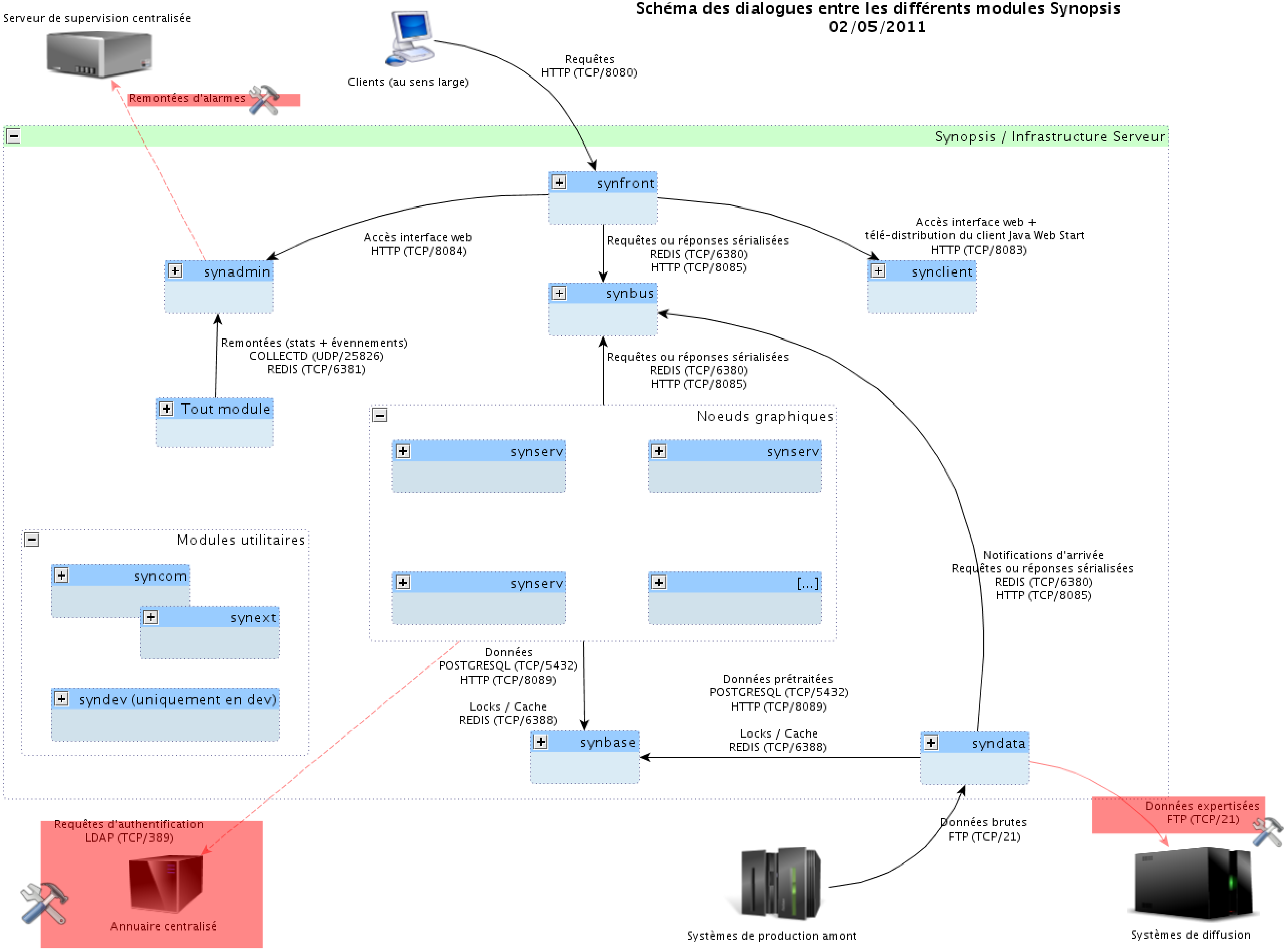
- *If you missed something...*
- *Technical and functional (new) specifications*
- *Technical foundations*
- *Diving into the server architecture*
- *Hardware infrastructure*

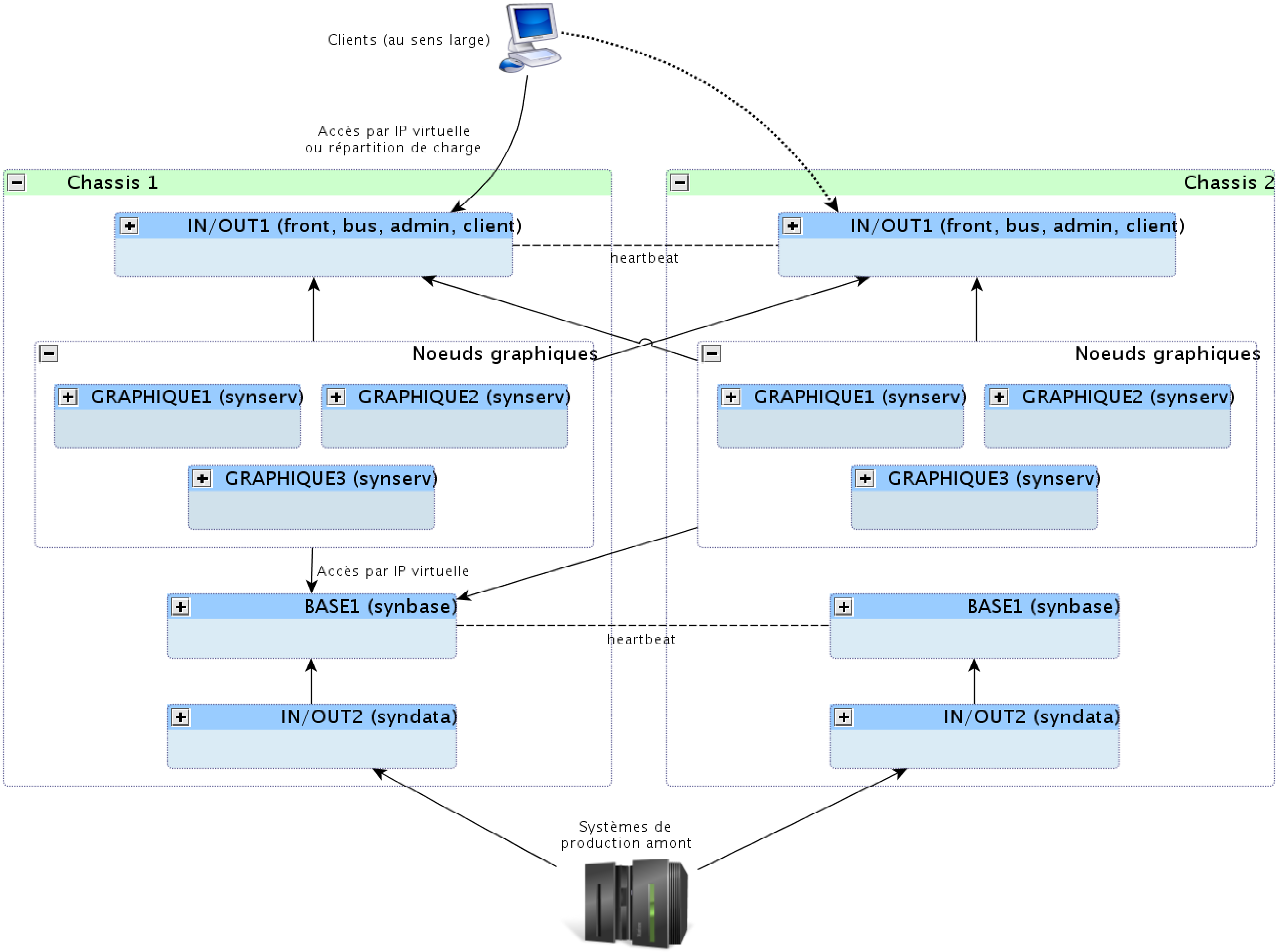
Technical foundations



"Synergie Next" Client architecture

Schéma des dialogues entre les différents modules Synopsis
02/05/2011




Clients (au sens large)
Accès par IP virtuelle
ou répartition de charge
Chassis 1
Chassis 2
IN/OUT1 (front, bus, admin, client)
IN/OUT1 (front, bus, admin, client)
heartbeat
Noeuds graphiques
Noeuds graphiques
GRAPHIQUE1 (synserv)
GRAPHIQUE2 (synserv)
GRAPHIQUE3 (synserv)
GRAPHIQUE1 (synserv)
GRAPHIQUE2 (synserv)
GRAPHIQUE3 (synserv)
Accès par IP virtuelle
BASE1 (synbase)
BASE1 (synbase)
heartbeat
IN/OUT2 (syndata)
IN/OUT2 (syndata)
Systèmes de
production amont

Diving into the server architecture

- Specifications to keep in mind :
  - Must work as standalone on commodity hardware behind a satellite receiver
  - Horizonal scalability
  - No service interruption upgrades
  - Must act as a cloud service :
    - Access from the LAN and from the Internet
    - No configuration or installation on the client side...
    - ... everything must be pushed from the server depending on the login used