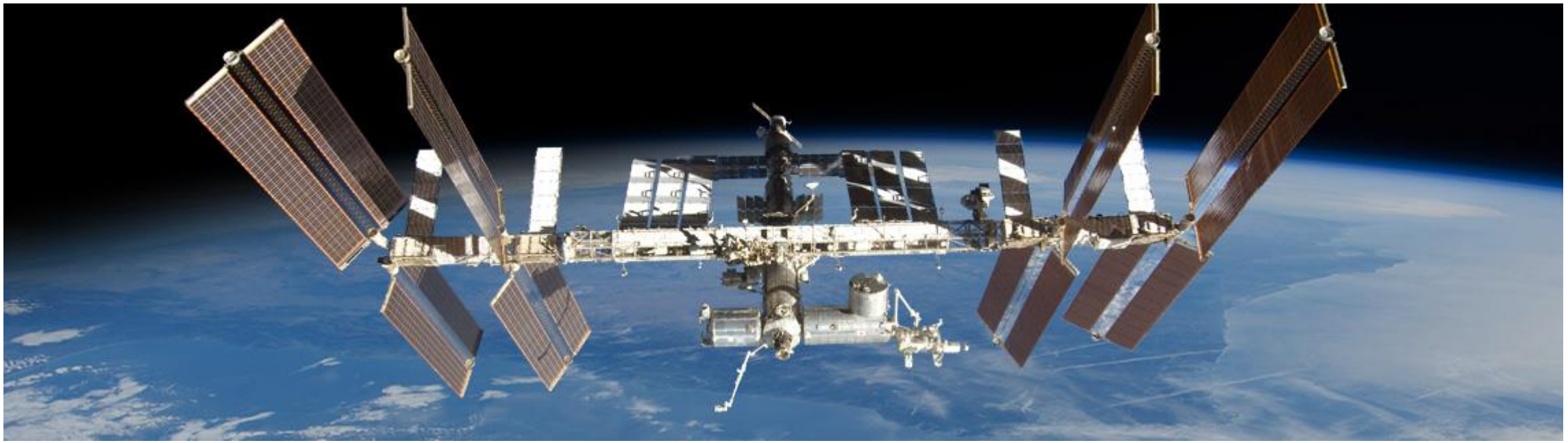
Architecture logicielle CDS

Première réunion – « ARCHI »
11 décembre 2018
P.Fernique



Buts : assurer l'adéquation et la pérennité de notre Système d'Information (SI)

- **Améliorer la connaissance** des uns et des autres sur le Système d'Information du CDS (grosse marge de progression possible)
- **Repérer les points fragiles**, ou à améliorer/transformer, puis mettre en œuvre ce qu'il faut pour **les résoudre**
- **Aider à la rationalisation, sécurisation, simplification, homogénéisation**, du-dit système
- **Si possible, éviter les fausses pistes** (techno inadaptée, mauvaise pérennité, …)

Comment appréhender le SI du CDS ? (aucun schéma/doc existants)

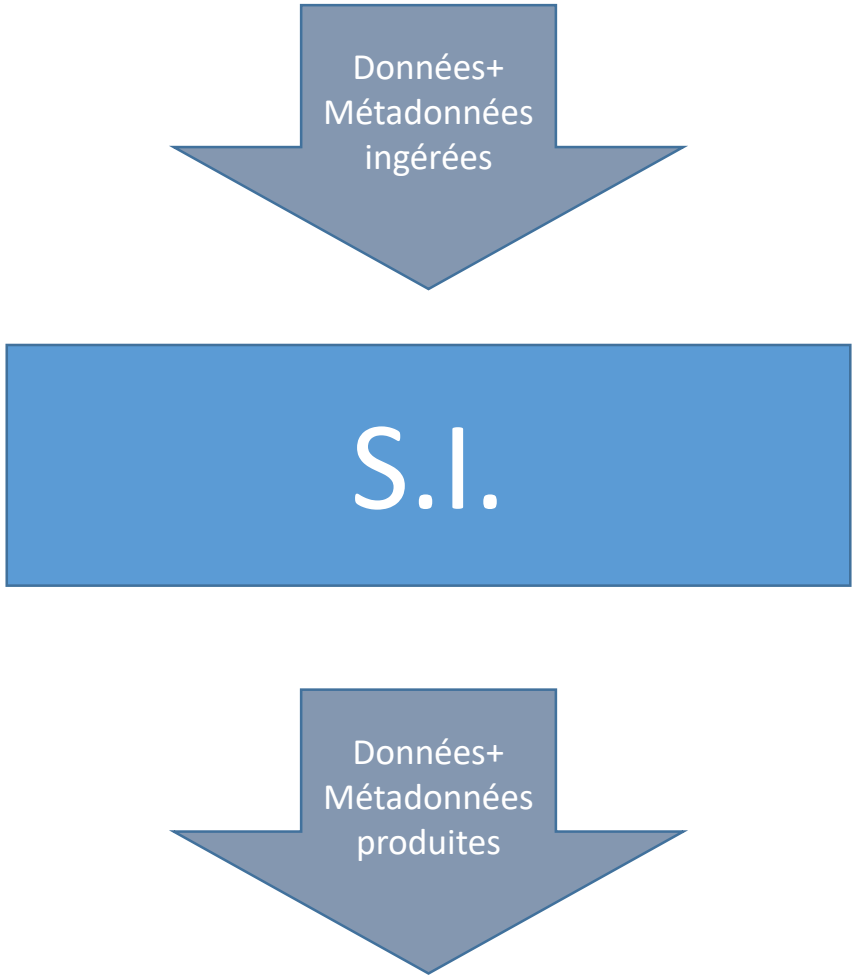
=> 1ère phase: interviews

=> Aujourd'hui: Conclusions (partielles) des interviews

(Attention: des erreurs ou imprécisions sont probables dans le schéma qui suit
=> il faudra que chacun vérifie ce qui le concerne)

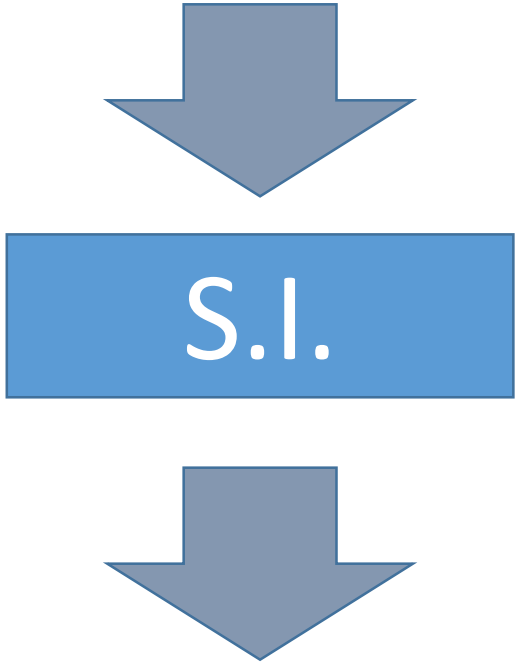
Anaïs, Soizick, Gilles, Cécile, FX, Thomas, Magali, Catherine, Pierre, François

et bientôt Bernd, Patricia…



Le SI: les éléments constitutifs

- **Les données/métadonnées d'entrée :** catalogues, articles, relevés pixels, …
- **Les données/données de sortie** : pages Web, VOTables/FITS/JSON de catalogue, HiPS…



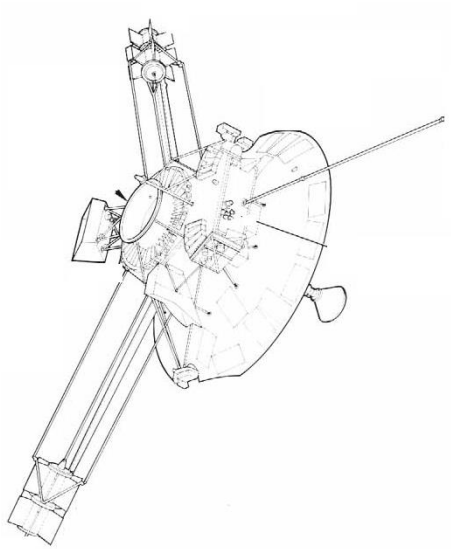
- **Nos moyens d'accès** à ces données/métadonnées
  - En entrée (ex: ftp, wget, dépôts automatiques, rsync…)
  - En sortie (ex: console, microfiche, formulaire WEB, script, TAP…)
- **Nos outils logiciels** (ex: DJIN, GSC4sim, anafile, cds.catfile ….)
- **Nos services** : modules, sous-modules (ex: Simbad, TAPVizieR…)
- **Nos conventions** (ex: bibcode, identifier data set (II/237/out), ..)
- **Nos sérialisations** (ex: parfile biblio, README, JSON.catana…)

- ***Nos procédures de traitement**: quel cheminement et valeur ajoutée?*
  => Propres à chaque équipe (Astronomes, DJINistes, VizieRistes, Cosimistes, Hipsistes….)
- ***Notre infrastructure hardware:** sur quel matériel ?*
  => Service informatique d'Obas à la manœuvre

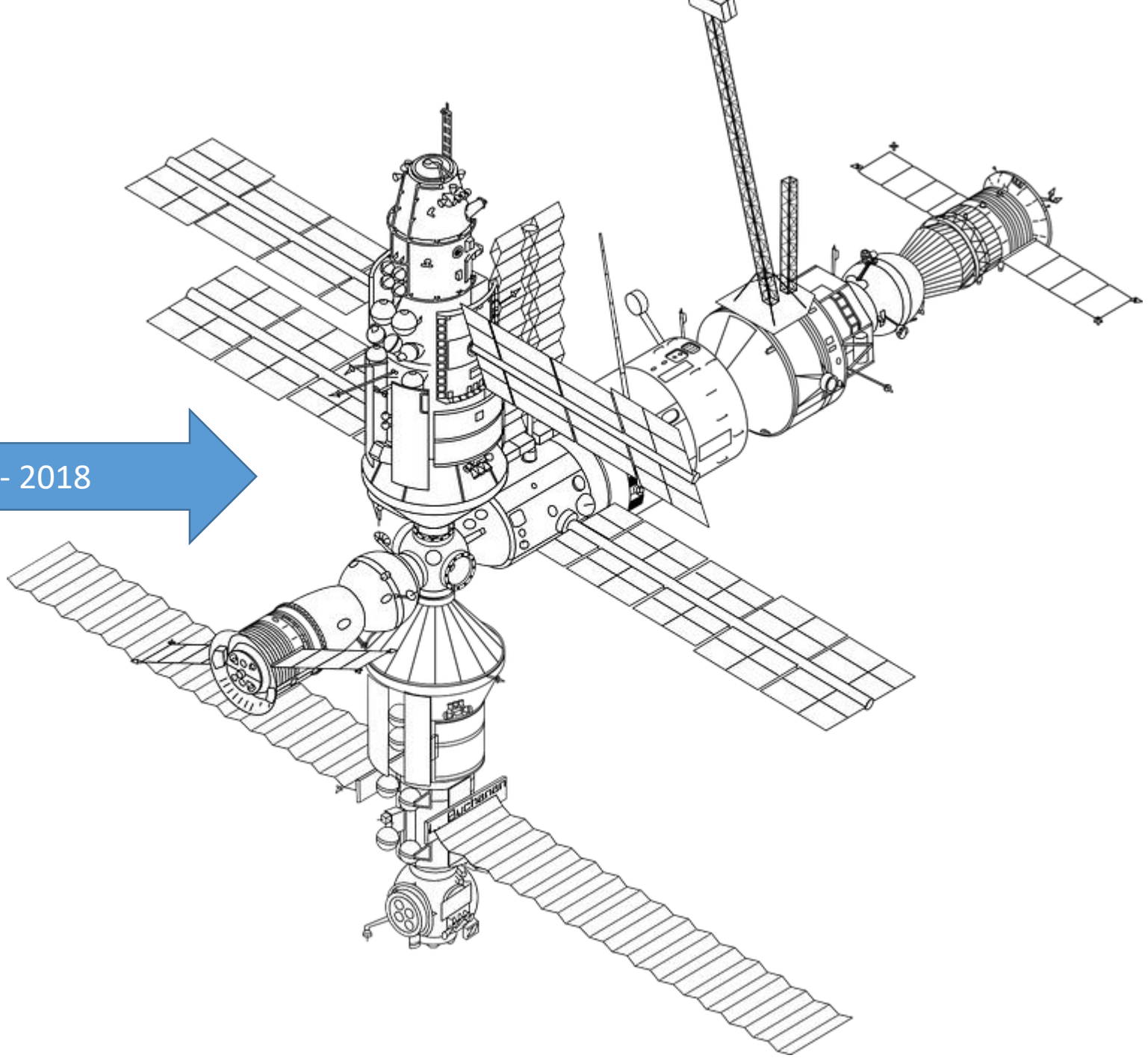
I) Les services du SI du CDS

Services & données résultats

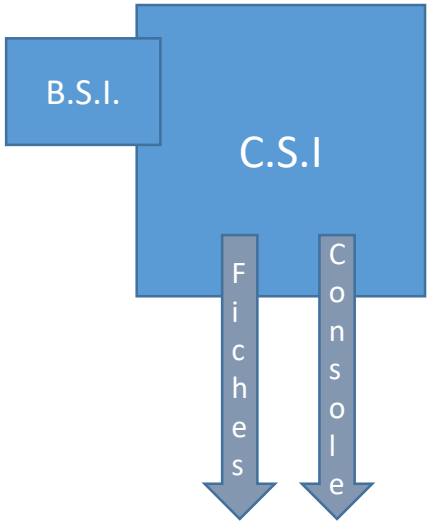
Evolution sur 45 ans….



1972 - 2018

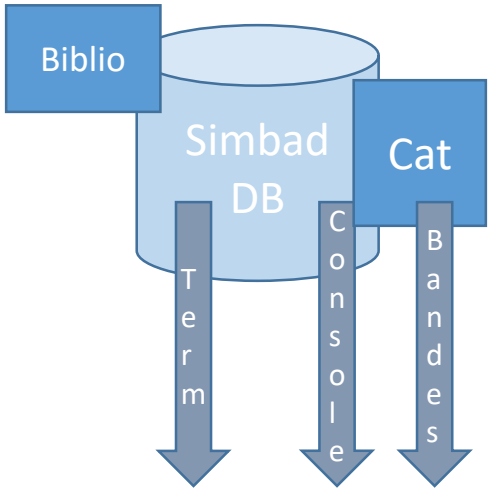


1972 – 1980 – CDS à Strasbourg
2 à 6 personnes



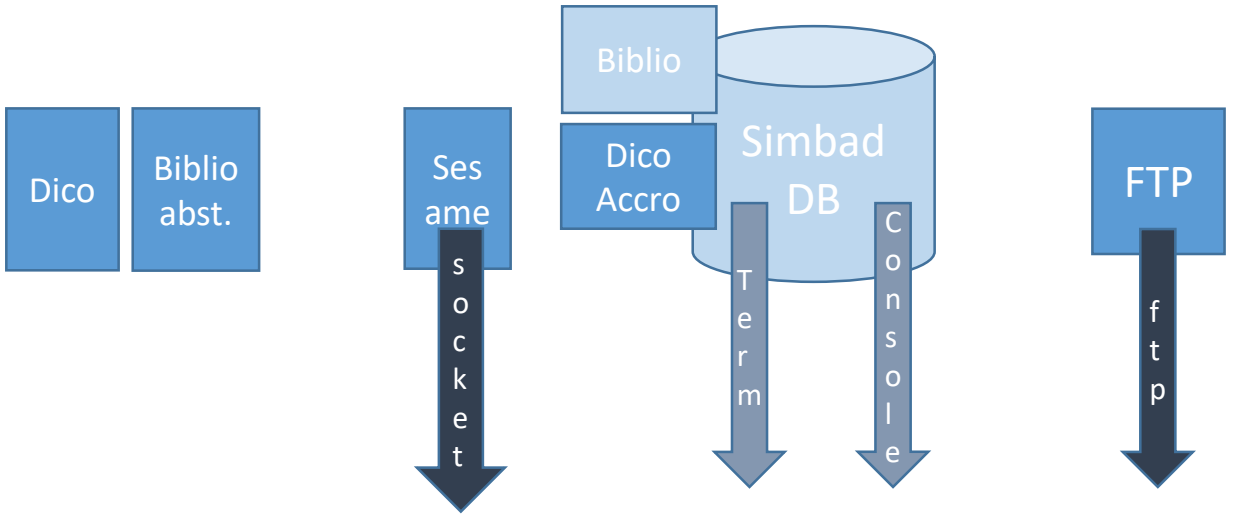
1 IBM 360/65 à
Meudon

1981-1985 – (83 -> towards galaxies)
6-12 personnes



1 Univac 1110 à Cronenbourg
Réseau dédié – lang PL1

1986-1994 – (91: +biblio, +catalogues)
12-20 personnes

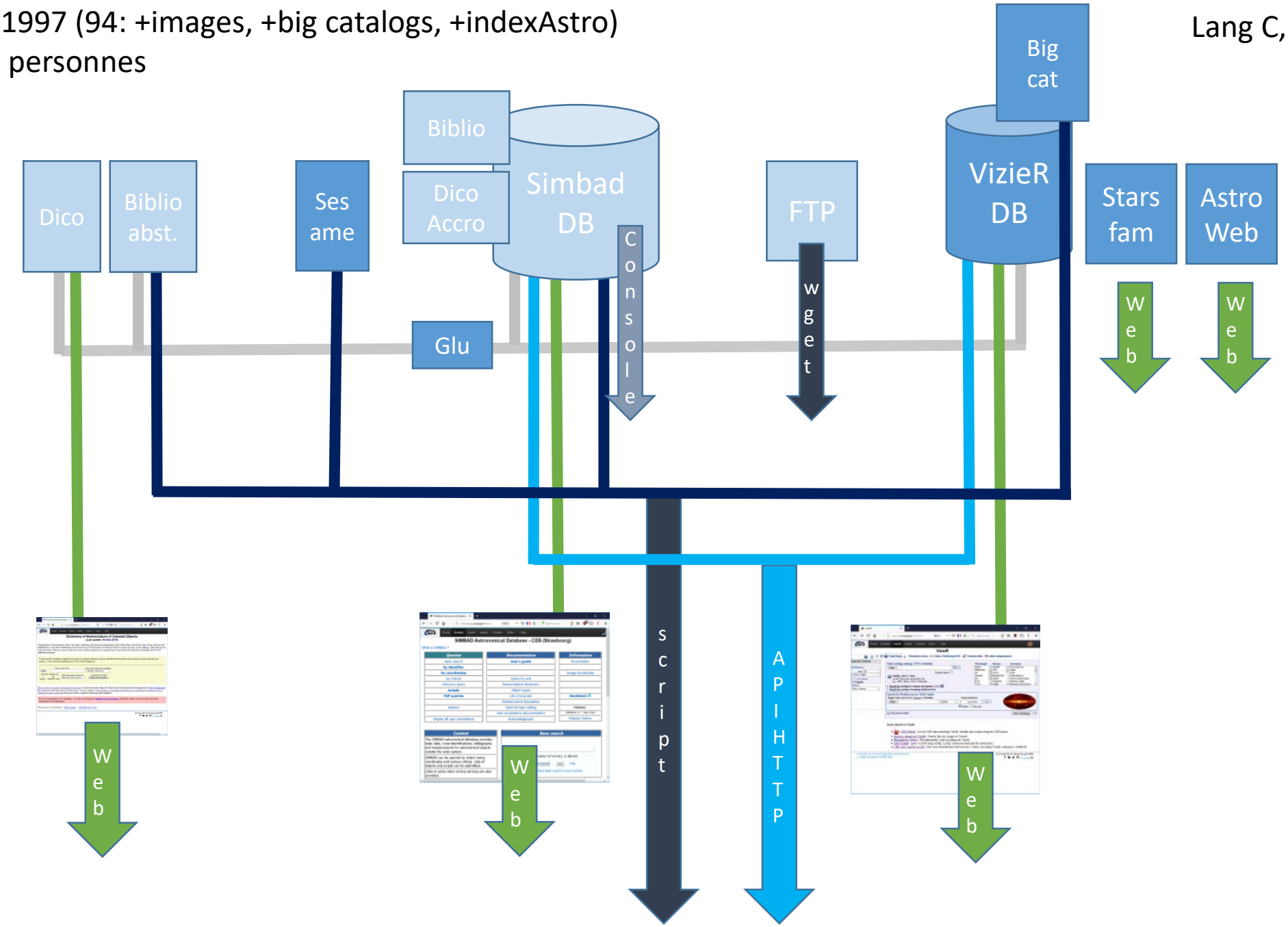


1 Vax/VMS, puis Dec/Ultrix à Strasbourg
lang C - DECnet,

puis 3 Suns
lang C, AWK – Internet

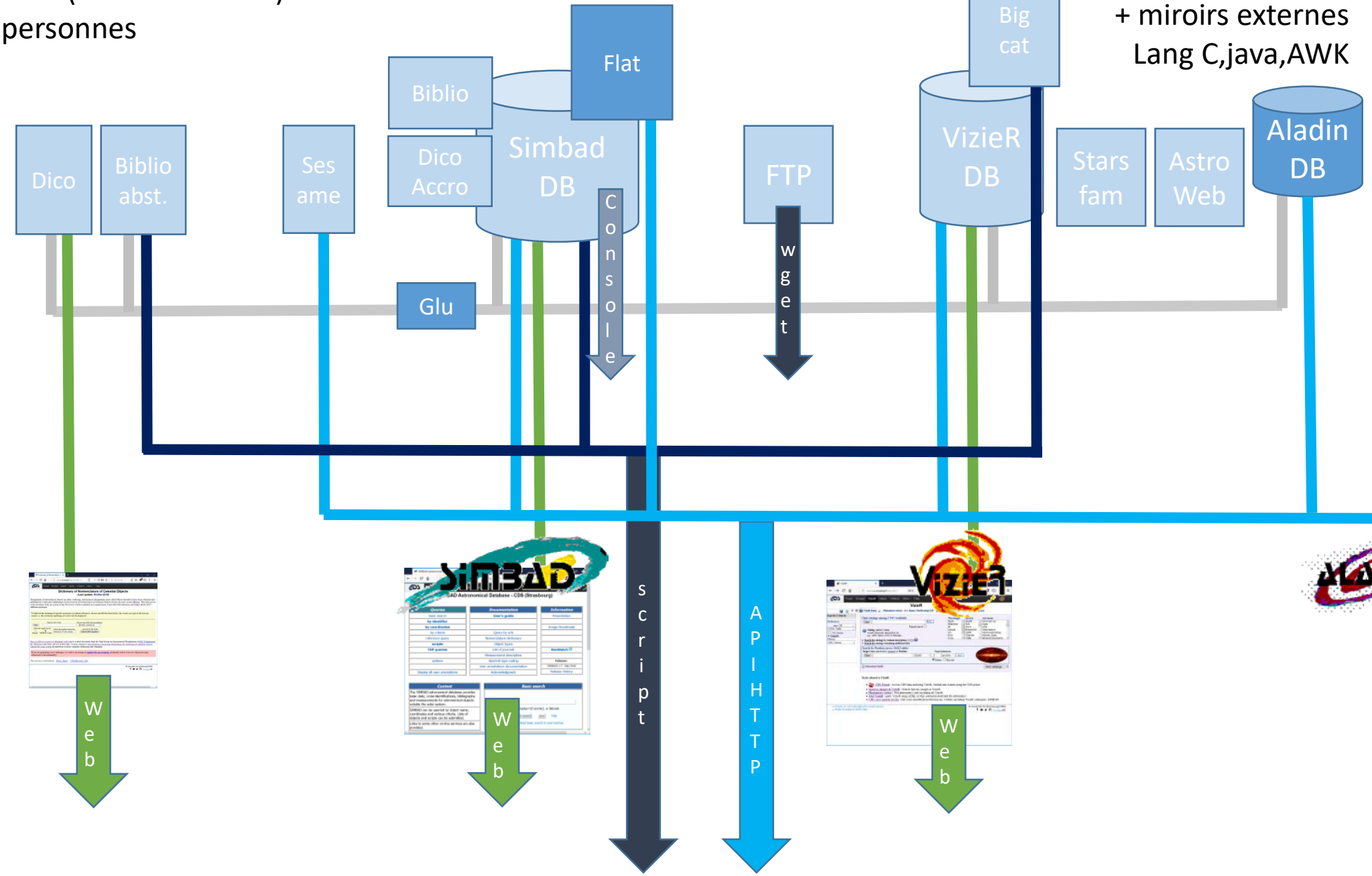
1995-1997 (94: +images, +big catalogs, +indexAstro)
20-25 personnes

5 Suns Obs
Lang C,C++,AWK,perl



1998-2007 (03: -indexAstro)
25-30 personnes

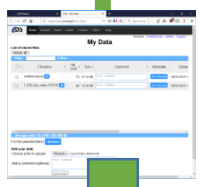
8 serveurs Obs
+ miroirs externes
Lang C,java,AWK

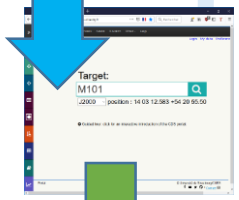


2009-2015 (12:-biblio, 14: +Big Images Surveys)
30-35 personnes

10 serveurs Obs + miroirs – lang: +python

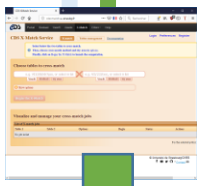
login
Dico
Biblio abst.
Ses ame
Biblio
Dico Accro
Flat
Simbad DB
Glu
console
FTP
wget
Big cat
VizieR DB
HiPS cat
Cat/X match
Aladi DB
Aladin HiPS images

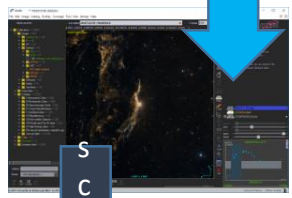

Web

Web


Web

Web
script
API HTTP

Web

Web

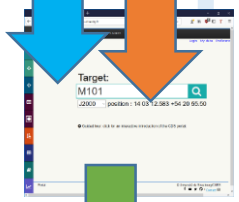
Web

script

2016-2018 (16: +AssocData)
40 personnes

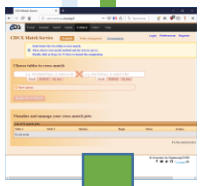
12 serveurs Obs + miroirs – lang: +python

login
Dico
Biblio abst.
Moc server
Ses ame
Biblio
Dico Accro
Simbad DB
Flat
Glu
console
FTP
wget
VizieR TAP
Ass. data
VizieR DB
Big cat
Cat/X match
HiPS cat
Aladi DB
Aladin HiPS images

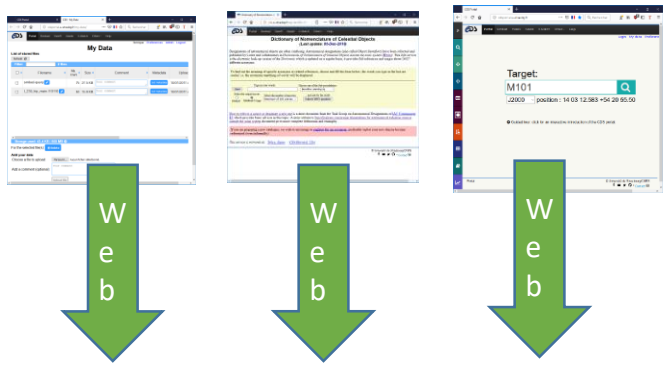


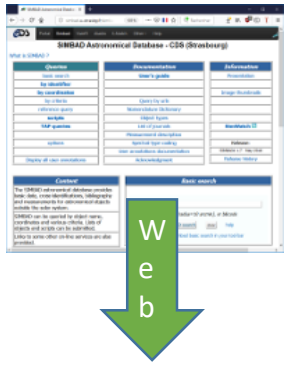


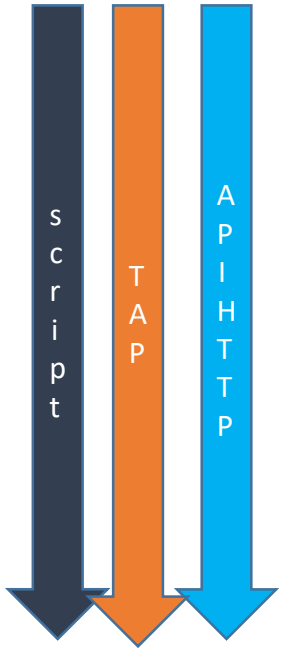


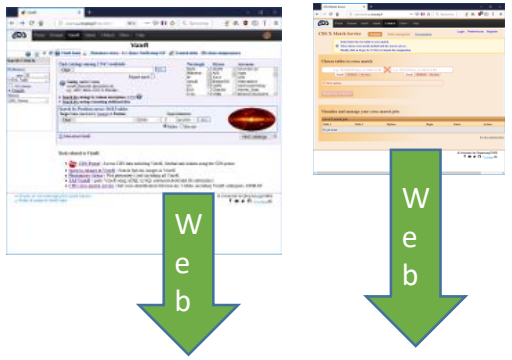


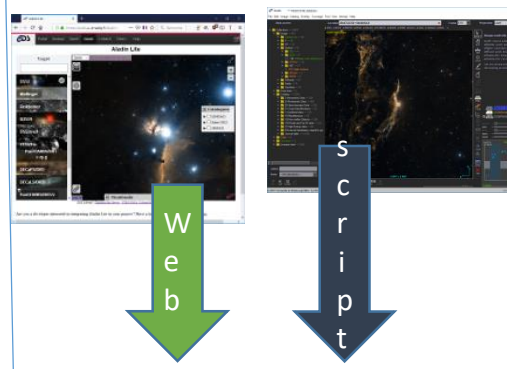
Web
Web
Web
Web
script
TAP
API HTTP
Web
Web
Web
script











Notre offre actuel

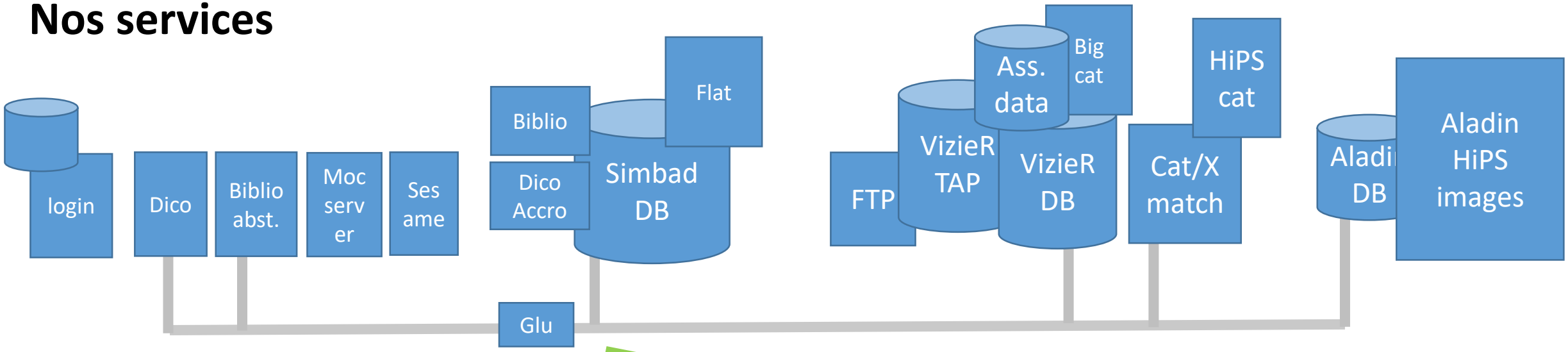
- 2 services majeurs Web (Simbad, VizieR)
  + des services Web annexes (xmatch,dico,doc,biblio,…)
- 2 « intégrateurs »: Aladin Desktop, le portail

- Des widgets (Aladin lite, SED, …)
- Des outils « scripts » (cdsclient/python)
- Des accès VO

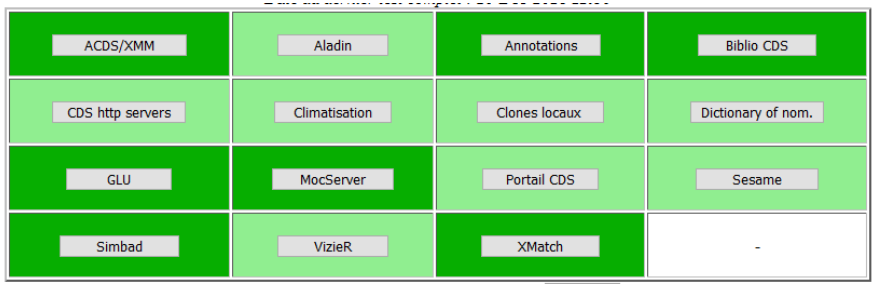
Mon point de vue: Notre offre est cohérente, diversifiée. Elle est remise régulièrement en question (ex: R&D python jupiter, appli android…) => ça marche plutôt bien

CENTRE DE DONNÉES ASTRONOMIQUES DE STRASBOURG

Nos services



- Une vingtaine de modules tout de même
- Inflation forte
- Une très bonne disponibilité >99% (ça marche plutôt bien)
- Quelques doublons
- Des éléments désormais peu utilisés

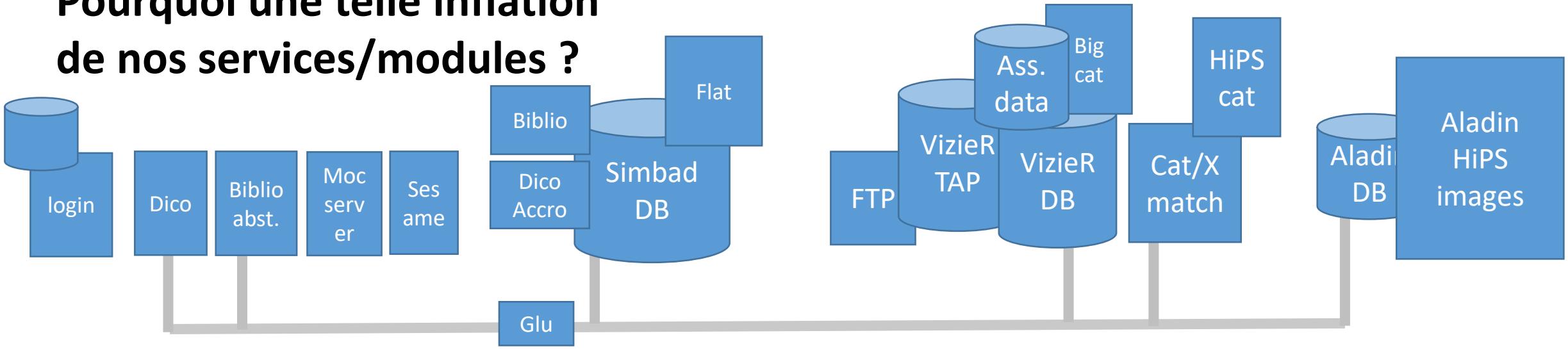


Mon point de vue: Nos services marchent bien, grâce notamment à une responsabilités/disponibilité des personnes en charge, à des miroirs, et à une supervision tatillonne + classeur bleu (un peu bricolage, mais ça marche)

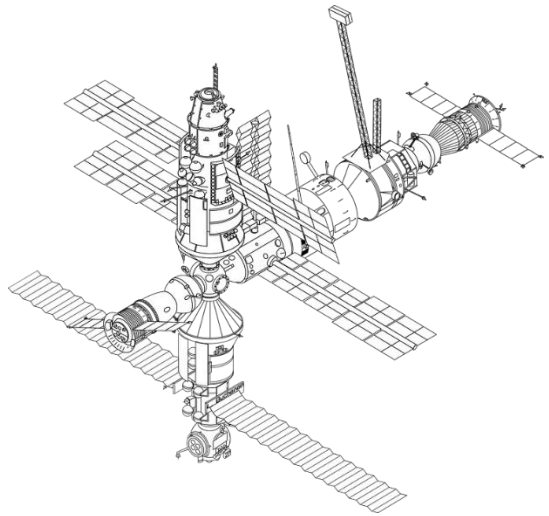
On pourrait sans doute simplifier ce qui pourrait l'être pour diminuer les dépendances en série, les doublons inutiles et la complexité de l'ensemble.
=> Mais requière de la disponibilité

- En revanche, gros déficit de doc, schémas globaux.
- Un meilleur système de stats aiderait aux décisions

Pourquoi une telle inflation de nos services/modules ?

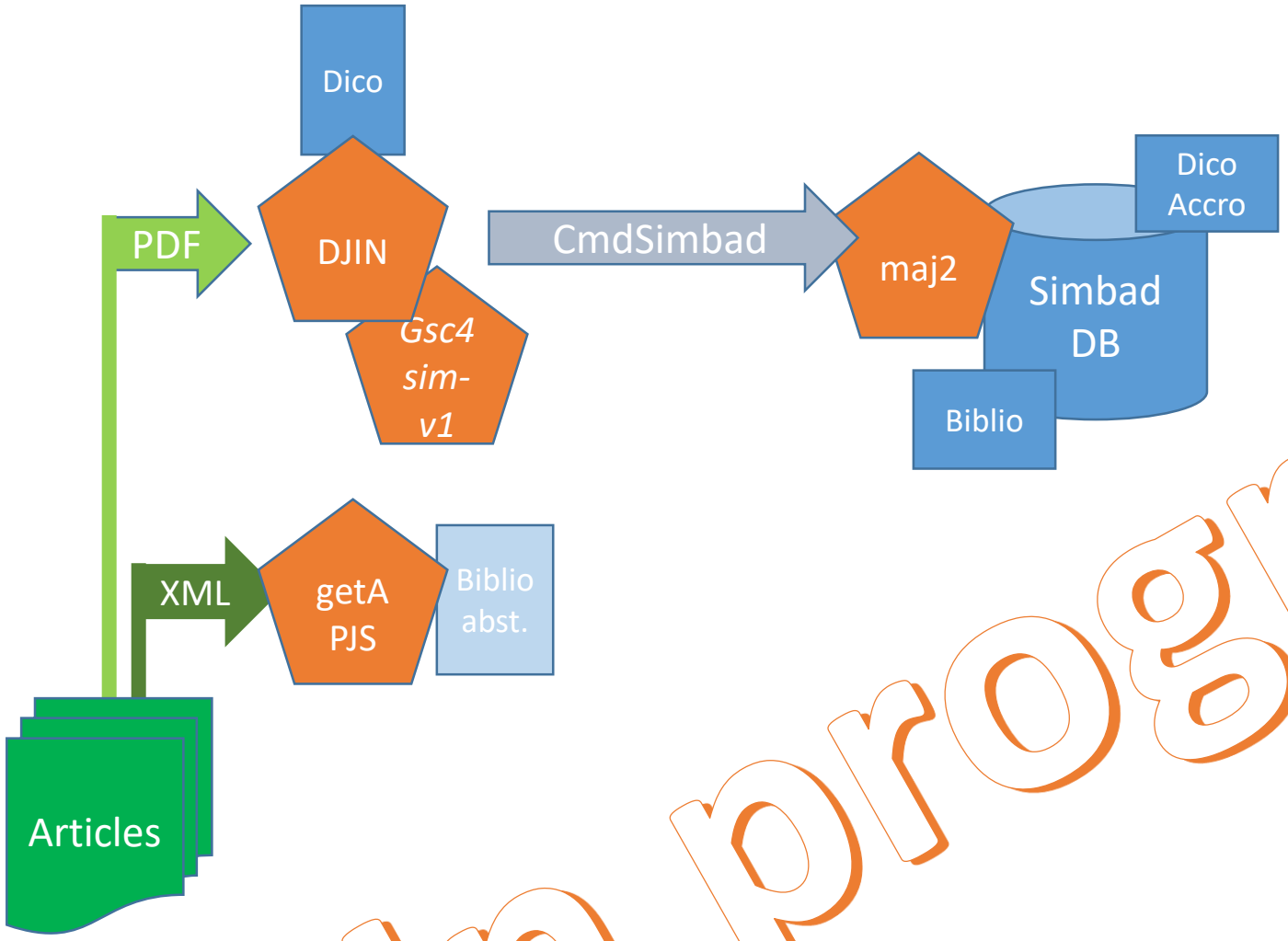


- **Hardware disponible**: nous pouvons techniquement nous permettre de multiplier les serveurs/machines (pas aussi simple avec un mainframe)
- **Contrainte technique** : Nouvelles contraintes de perf non supportables par le module initial (ex: Big cat, xmatch, Simbad flat,…)
- Des services relativement disjoints reflets de notre **organigramme**
- **Facteur organisationnel**: R&D (souvent un greffon) puis mis en opérationnel (Dico, HiPS, MocServer, …)
- **Contrainte temporelle**: Pas le temps de supprimer les « anciens » modules même si leurs raisons initiales n'existent plus (ex: Biblio abst, Dico pourraient être réintégré dans Simbad…)
- Eléments liants (Glu, MocServer, Sesame, …)



II) Le services du Système d'Information du CDS

Traitement des données et métadonnées ingérées (état 2018)



In progress...

En cours d'établissement, nécessite un peu plus de temps que ce dont je disposais
=> A voir pour les prochaines réunions…

gsc4sim	Code *François O.*, Gilles coté Vizier, Anaïs coté Simbad	Shell, Awk, aclient…	Simbad	Code d'interrogation de Vizier pour récupérer les champs et autres data pour un nouvel objet à insérer dans Simbad. Génère les commandes maj2 pour Simbad (majSimbad) => Soucis de maintenance (bugs)
DJIN	Code *Christian Bonin*	Java / Swing + lib pdf (pdfbox 0.8)	Simbad	Répérage des noms d'objets à partir des articles PDF, puis génération des commandes Simbad correspondantes => en cours de remplacement
Cosim	Code Anaïs	Java (java-FX)	Simbad	Ex-raccord : génère les commandes de maj Simbad en fonction d'une table parfile
maj2	Code Anaïs	Java (jdbc)	Simbad	Maj de Simbad à partir d'un fichier de commandes de maj
cgiprint	*Code Fox*	C	La plupart des services concernés	Conversion Latex en HTML avec prise en compte des macros VizieR => Code indispensable,
cds.catana	Code FX	Java	FX bigcat, VizieR	Génération metadata/stats pour grands catalogues
API.SAADA	Code Laurent Michel	Java	VizieR	Ingestion des données FITS
anafile	*Code Fox,* géré par Gilles	C	VizieR	Lecture d'une table ASCII

III) Nos outils internes

- Une **vingtaine** de logiciels/packs/outils
- En **C & java** pour la plupart,
- Qq autres en JS, python, perl, AWK et shell
- La plupart **développés par le CDS**
- Qq uns par des CDD de passage
- Un bonne part reste du code Fox (gsc4sim, findXXX, cgiprint, aclient/aserver, anafile, astropos, jdate, acut,…)

Mon point de vue: Nous tournons encore en grande partie sur notre élan… beaucoup de codes ne sont pas/plus sous contrôle (ex: cgiprint), ou sont à améliorer (ex: DJIN, cds.catfile, …)
=> Malgré beaucoup d'efforts, encore une grande fragilité, notamment en cas de migration impossible, ou d'un départ de la personne en charge

IV) Conventions CDS

Bibcode	CDS/NED créé en 1983 repris par ADS	Identification (lisible) d'une référence biblio => concurrence/cohabitation avec DOI
Clés Parfile	CDS (Fox)	Identification (une lettre assez cryptique) d'un champ spécifique (titre, coord, abstract, acronyme…). => Nomenclature interne CDS, presque cohérente à travers tous les outils, mais pas tout à fait
Identificateur Tables	CDS/ADS (Fox)	Identification (lisible et hiérarchique) d'une table => plusieurs dérivées incohérents (préfixe « CDS/ » (MocServer), prefixe « vizier : » (Xmatch), suffixe « ?xxxx » (VizieR), préfixe « CDS.vizier / » (VO registry))
Identificateur type d'objet	CDS (Fox & Marc) => en cours de reprise par Cécile & Anaïs	Identification hiérarchique (lisible) de tous les types d'objets - Pour le moment limité à 4 niveaux. - Ne décrit pas le graphe (uniquement un arbre du graphe) - Une alternative sur 4 octets (à la IP) existe - Nouvelle numérotation Cécile (simple numéro d'ordre) => comment gérer les ajouts/modifs ?
Tag GLU	CDS (Pierre)	Identification d'URLs => Délaissé par les nouveaux modules issus de R&D récentes (ex : xmatch, portail)

Demi-douzaine de systèmes d'identification des objets manipulés

Mon point de vue: indispensable pour la construction de notre SI. Ce sont les briques de base. La plus grande attention doit y être apporté.

- Difficulté d'établissement/évolution notamment pour les types d'objets
- Des divergences sur les identificateurs des tables et dataset.
- Certaines de nos conventions de longues dates sont désormais plus facilement délaissées (notamment clés parfile)

=> A faire converger asap

V) Sérialisations CDS

README	CDS (Fox)	VizieR	Descriptions des méta données d'un catalogue en format ASCII éditable vi
Parfile biblio	CDS (Fox)	Biblio, Simbad	Données et métadonnées de références bibliographiques (ASCII éditable vi)
Parfile Simbad	CDS (Fox)	Oldsim	Données et métadonnées des enregistrements Simbad (ASCII éditable vi)
Parfile GLU	CDS (Pierre)	Glu	Metadonnées et URL des services
majSimbad	Historique CDS	Maj2 -> Simbad	Liste de commandes de maj de Simbad (une comm[illegible]de [illegible]gne, directive une lettre, suivi de para[illegible]tre

Extrait

rcf	CDS FX	[illegible] FX (java)	Fichier binaire pour grands catalogues
Table binaire	Propre à VizieR	Fox (en C)	Fichier binaire pour les grands catalogues (méthode Fox)
.status	Propre à VizieR	VizieR	Listes de macros latex décrivant les étapes d'ingestion dans VizieR pour un catalogue donnée
CSV naïf	FX code	Grand catalogue	Séparateur «, » + une ligne de header

- Le CDS est grand inventeur & consommateur de sérialisations « maisons ». La plupart sont parfaitement adaptés à nos besoins (pérennité, performance, simplicité…)

Mon point de vue: Les sérialisations entrent dans une grande part dans la solidité de notre SI.
la réutilisation d'une sérialisation pré-existence est toujours préférable à ré-inventer la roue, et encore mieux si elle est standardisée (ex: MOC, HiPS, VOTable). Un facteur très important pour la simplification de notre SI.
Malheureusement pas toujours possible (évolution des données qui n'entrent plus-ex: unicode), pas toujours adéquate (ex: rcf)

Parfile est au cœur du SI et pourtant délaissé:
Suggestion 1*:* Faire évoluer le parfile pour supporter l'unicode ;
Suggestion 2: Se définir une alternative du parfile en JSON (en gardant la sémantique des clés) ?

Pour la suite…

Méthode que je souhaite proposer

- **Une réunion « ARCHI » tous les 2 mois** (environs) pour faire le point sur le SI. Tour de table des changements, et des problèmes. Etat d'avancement des « **points chauds** » en cours de résolution. Repérage des nouveaux « points chauds ».
- Qui : les responsables techniques et scientifiques de chaque service + quelques personnes clés volontaires (péréquation efficacité/nombre ?)
- Détermination des priorités, éventuellement tranchée par Mark
- Mise en place de **groupes ad-hoc** pour les « points chauds » à traiter en priorité (pas trop à la fois en fonction des forces disponibles)
  => On commence tout de suite -> DJIN

Pourquoi maintenant un tel rôle ?

- La plupart des briques du Système d'Information du CDS a été bâti par François O et Marc W. Ils sont désormais retraités depuis 2 ans. Cette connaissance a été transmise, mais au sein de chaque service (Simbad, VizieR…). La connaissance globale et la cohérence de l'ensemble doit être assuré.
- Or nous sommes désormais bien plus nombreux (en personnes et en services) => la solution autour d'un café est un peu limite.
- Actuellement, nous roulons plutôt bien, et nous sommes reconnus pour cela. Mais nous vivons sur notre lancé (le SI de nos prédécesseurs est solide). Ca ne peut pas durer éternellement.

Ecueils à éviter

- Ne surtout pas freiner les initiatives (proto, R&D, …) propres à chaque service (ex: DJIN pour Simbad, HiPS pour Aladin, nouvelles pages Web).
- Eviter de faire glisser les responsabilités propres à chaque service sur ce nouveau rôle. Idem pour l'infra.
- Maintenir à jour la connaissance du SI : Comment éviter le document « usine à gaz » toujours en retard d'une guerre ?
- Etre efficace de la manière la plus légère possible (si c'est juste pour une réunion en plus….)
- Ne pas TOUT remettre sur le tapis (y aller étape par étape)

Réactions, commentaires ?

A noter, 2 réunions « ad-hoc » déjà prévues

- GSC4sim
- DJIN