

Web Services sur IBM i

9 et 16 Novembre 2017



@GaiaFrance

www.gaia.fr
www.know400.fr

contact@gaia.fr

Agenda

- Horaires

- 8h45 -> 9h15 : accueil / petit déjeuner
- 9h15 -> 9h30 : news
- 9h30 -> 12h30 : présentation et échange

- Plan

- Les web services SOAP et REST, usage pour l'entreprise
- Fonctionnement IWS client et serveur
- Les technologies sous-jacentes
- Possibilités et limites
- L'administration et l'exploitation des serveurs IWS
- La sécurité
- SQL en tant que client ?
- Nos projets clients, retour d'expérience





News !

- Disponibilité de RDi 9.6
 - 21 novembre 2017
 - Fonctionnalités
 - Modifications techniques : Eclipse 4.6.3
 - Visualiser facilement les variables non utilisées, les messages de compilation directement dans le source sous forme d'annotations
 - Lancer la fonction de couverture de code à partir d'une ligne de commande (pour intégration dans un processus de build)
- Fin de support
 - Apache 2.2 délivrée en IBM i 7.1
 - Java 6
 - 31 décembre 2017
- Fin de support annoncé pour 7.1
 - 30 avril 2018

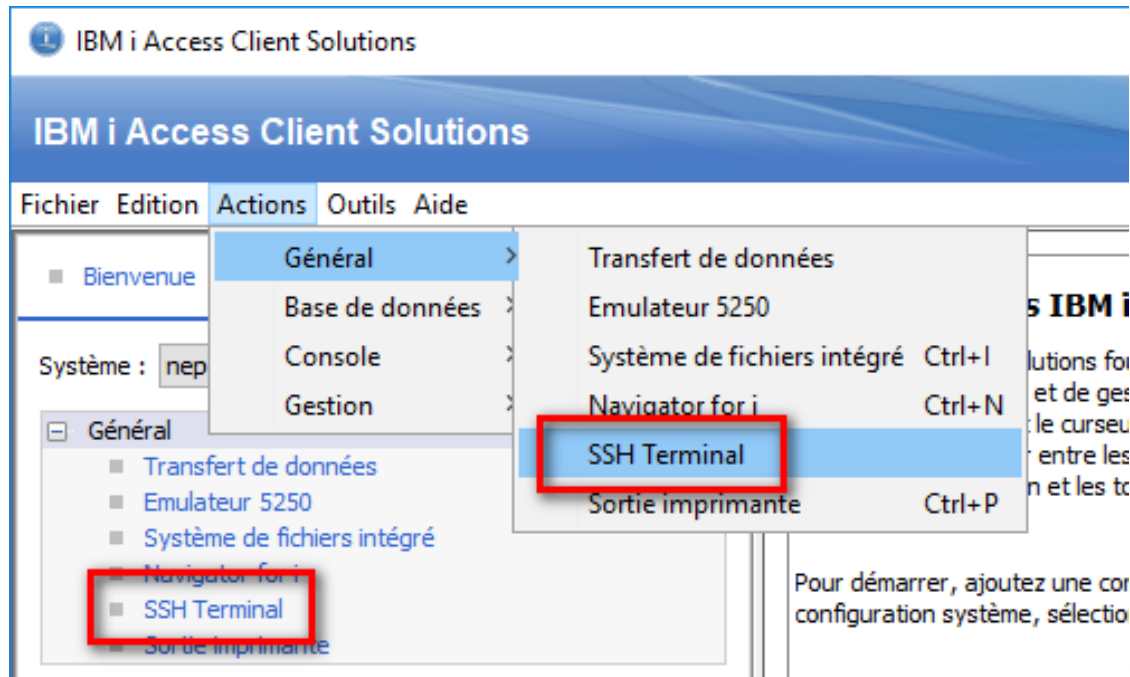
GAIA

News !

- 7.2 TR7 / 7.3 TR3
 - Disponible au 27/10/2017
 - 8/12/2017 pour les évolutions DB2 Web Query
 - Annonce officielle
 - http://www-01.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/8/877/ENUSZP17-0628/index.html&lang=en&request_locale=en
 - Documentation
 - <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/IBM%20i%207.3%20%20TR3%20Enhancements>
 - Principales fonctionnalités
 - Publication JSON
 - Services DB2 for i : nouveaux et modifiés
 - Support hardware
 - Produits open source
 - **Web service serveur : appel du programme exposé sur un autre IBM i**

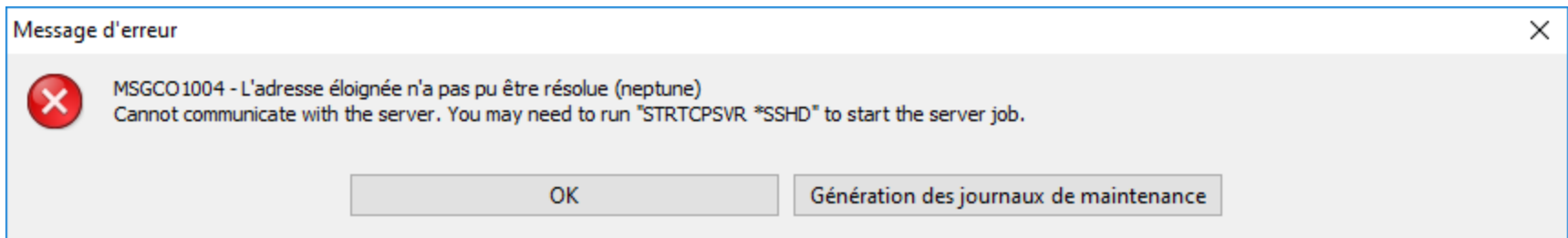
ACS - SSH

- Option disponible depuis la 1.1.7.2 du 22/10/2017
 - Permet de lancer une session SSH, via un outil déjà disponible sur le poste client
 - Accessible depuis

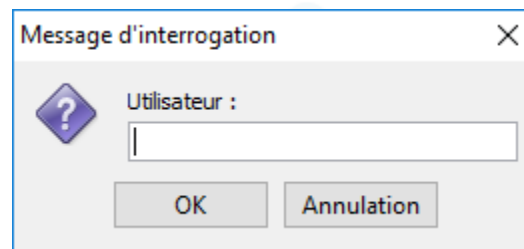


ACS – SSH : Exécution

- Le serveur SSH doit être démarré sur le serveur
 - STRTCPSVR *SSHD

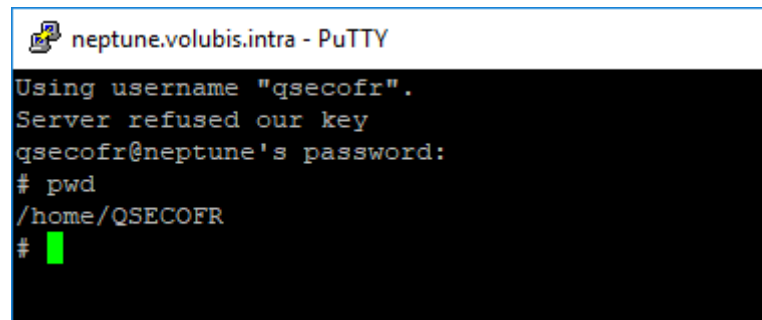


- Lancement du gestionnaire SSH déclaré dans le fichier de configuration
 - Dès lors qu'une clé personnelle a été importée, l'outil demande le profil utilisateur à connecter



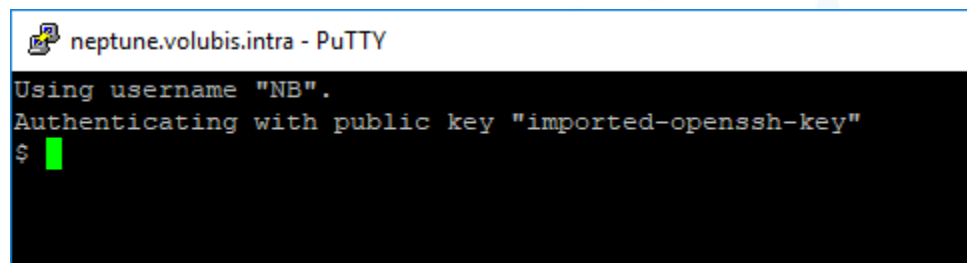
ACS – SSH : Session

- Si l'utilisateur ne dispose pas de clé déclarée
 - Demande du mot de passe



```
neptune.volubis.intra - PuTTY
Using username "qsecofr".
Server refused our key
qsecofr@neptune's password:
# pwd
/home/QSECOFR
# █
```

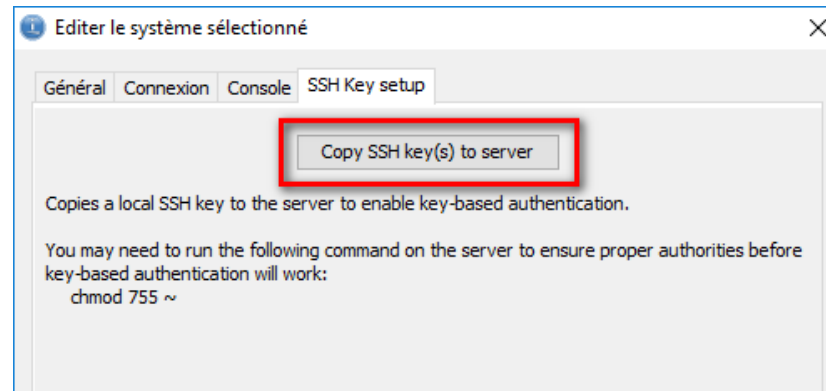
- Sinon, accès direct à la ligne de commande



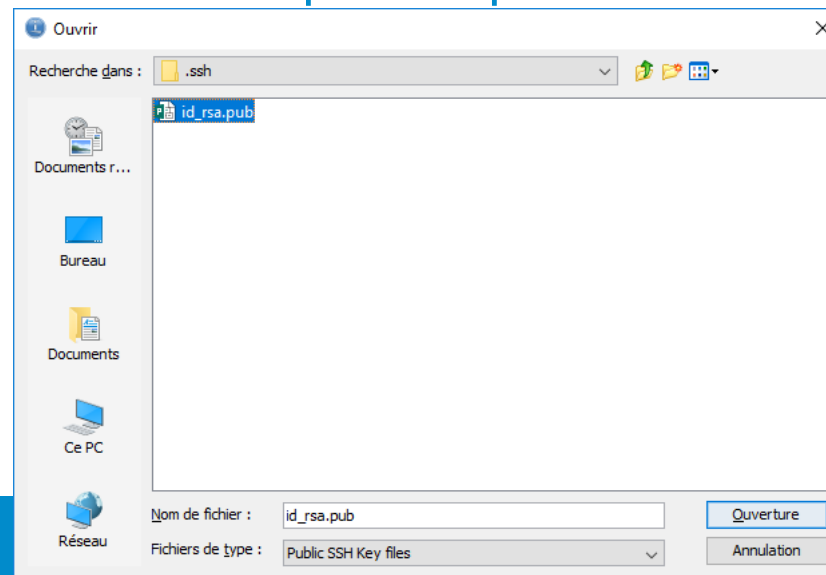
```
neptune.volubis.intra - PuTTY
Using username "NB".
Authenticating with public key "imported-openssh-key"
$ █
```


ACS – SSH : Importer une clé

- Dans la configuration des systèmes

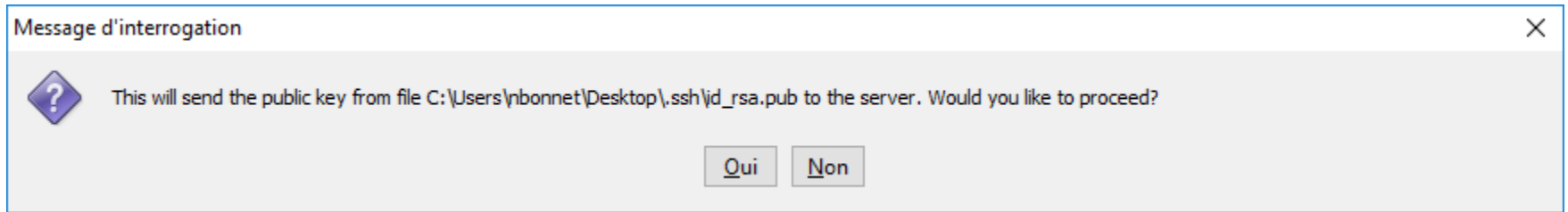


- Sélectionner votre clé publique via la boîte de dialogue

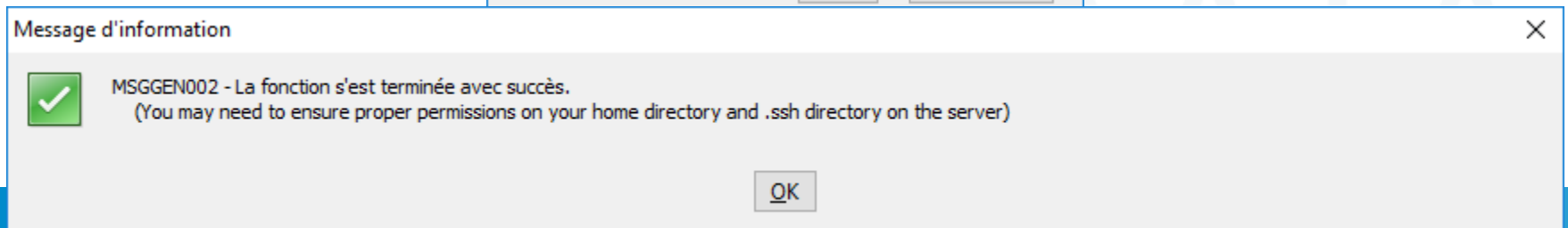
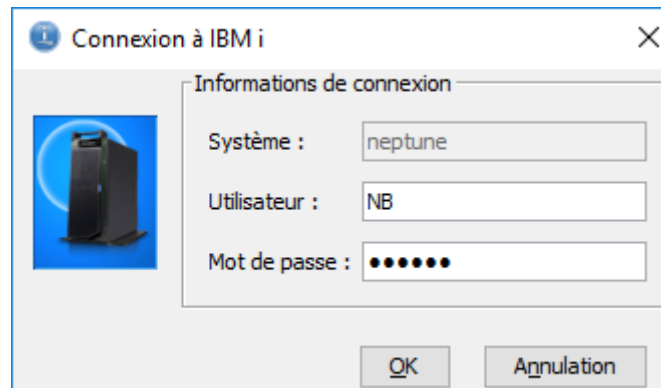


ACS – SSH : Importer une clé

- Confirmation



- Cela nécessite une connexion au serveur



Prochains événements

Date 2017	Evènement
15 Novembre	Common Europe iTour sur le thème « GDPR et IBM i » 1/2 journée 14h00-17h30 - IBM Client Center de Bois-Colombes
16 Novembre	Webinar sur le thème « Gestion du cycle de vie des applications » 10h00-11h00, animé par ITHEIS et IBM
20-22 Novembre	Workshop sur le thème « IBM i Security & GDPR » réservé aux partenaires IBM et à leurs clients - BM Client Center de Montpellier
28 Novembre	Webinar sur le thème « IBM i et RGPD » 10h00-11h00, animé par TRADER'S et IBM
7 Décembre	POT (Proof Of Technology) « Anonymisation des données sensibles de DB2 for i dans le cadre de la GDPR » IBM et Ithéis, à IBM Bois-Colombes

Date 2018	Evènement
6 Février	Evènement sur le thème de la GDPR Ouvert à tous avec IBM, ITHEIS et CILASOFT
17-18 Mai	L'Université IBM i : l'évènement annuel dédié à l'IBM i où l'on fêtera les 30 ans de la plateforme !

Web Services

Définitions et usages



Service Web

- Un Service Web permet l'appel d'un programme, ou d'une procédure
 - C'est un programme
 - Qui interagit avec d'autres programmes
 - En utilisant les protocoles les plus standards
- Ce n'est pas
 - Un site web ou une webapp : ne possède pas d'interface utilisateur donc ne peut être appelé que depuis une application
- Objectifs
 - Interopérabilité
 - Des programmes
 - Des plateformes



Service Web

- Un air de déjà vu ?
 - RMI, CORBA, DCOM
 - Des technologies
 - complexes issues de RPC (Remote Procedure Call)
 - Fortement couplées
 - Difficilement interopérables
- Les garants de l'interopérabilité
 - Des protocoles construits sur les standards du web !
 - HTTP/S, URI, XML / JSON
- Avantages
 - Transite facilement sur les réseaux (HTTP/S port 80/443)
 - Existe sur tous les systèmes
 - Couplage lâche



Architecture

- Les Services Web
 - Permettent de connecter des applications
 - Couplage lâche
 - Chaque composant de l'application
 - A un rôle défini
 - Échange des données avec d'autres composants (appels de service), dans un cadre défini et commun à tous
 - Exemple : la prise électrique !
 - Par opposition, un couplage fort donne le fameux « plat de spaghettis » (dépendance technique et fonctionnelle)
 - Non testable, non réutilisable
- Sont une des implémentations SOA (Service Oriented Architecture)

Scénarios

- Avec des partenaires
 - Echange de documents – facturation
 - Calcul de taxe
- Avec des inconnus
 - Affichage de la météo au login (pour le plaisir)
 - Google
- Entre nos applications
 - Connexion d'applications
 - ODI
 - SAGE X3
 - Salesforce



Les web services SOAP et REST

- Services Web de type REST
 - Basé sur l'architecture Web : HTTP, URI
 - Les données en entrée sont encodées soit l'URL, soit en JSON ou XML dans la requête
 - Les données en sortie sont encodées en JSON ou en XML
 - Standards d'interopérabilité en cours : Swagger / WADL
- Services Web de type WS-^{*}
 - Standards supplémentaires mis en œuvre
 - SOAP (Simple Object Access Protocol)
 - WSDL (Web Service Description Language)
 - UDDI (Universal Description Discovery and Integration)
 - Les données sont encodées en XML

Les web services SOAP et REST

POST http://neptune:10054/web/services/bankService/bank HTTP/1.1

Accept-Encoding: gzip,deflate

Content-Type: application/soap+xml;charset=UTF-8

Content-Length: 344

Host: neptune:10054

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:bank="http://bank.univ2016/">
  <soap:Header/>
  <soap:Body>
    <bank:getoperationarray>
      <arg0>
        <p_ID>1</p_ID>
        <p_MAXRESULT>20</p_MAXRESULT>
      </arg0>
    </bank:getoperationarray>
  </soap:Body>
</soap:Envelope>
```

GET http://neptune:10054/web/services/bankrest/operations/1/20 HTTP/1.1

Accept-Encoding: gzip,deflate

Host: neptune:10054

Usage pour l'entreprise

- Interconnecter les applications
 - Internes à l'entreprise
 - Site web marchand <-> back office
 - Application mobile <-> back office
 - Avec des fournisseurs
 - Contrôle d'adresses postales
 - Récupération de coordonnées GPS depuis une adresse postale
 - Cours de la bourse en temps réel
 - Avec des partenaires
 - Gestion pour compte
 - Avec des institutions
 - Open Data
 - Administrations, l'Union Européenne
 - Avec des clients

GAIA

IWS serveur et client



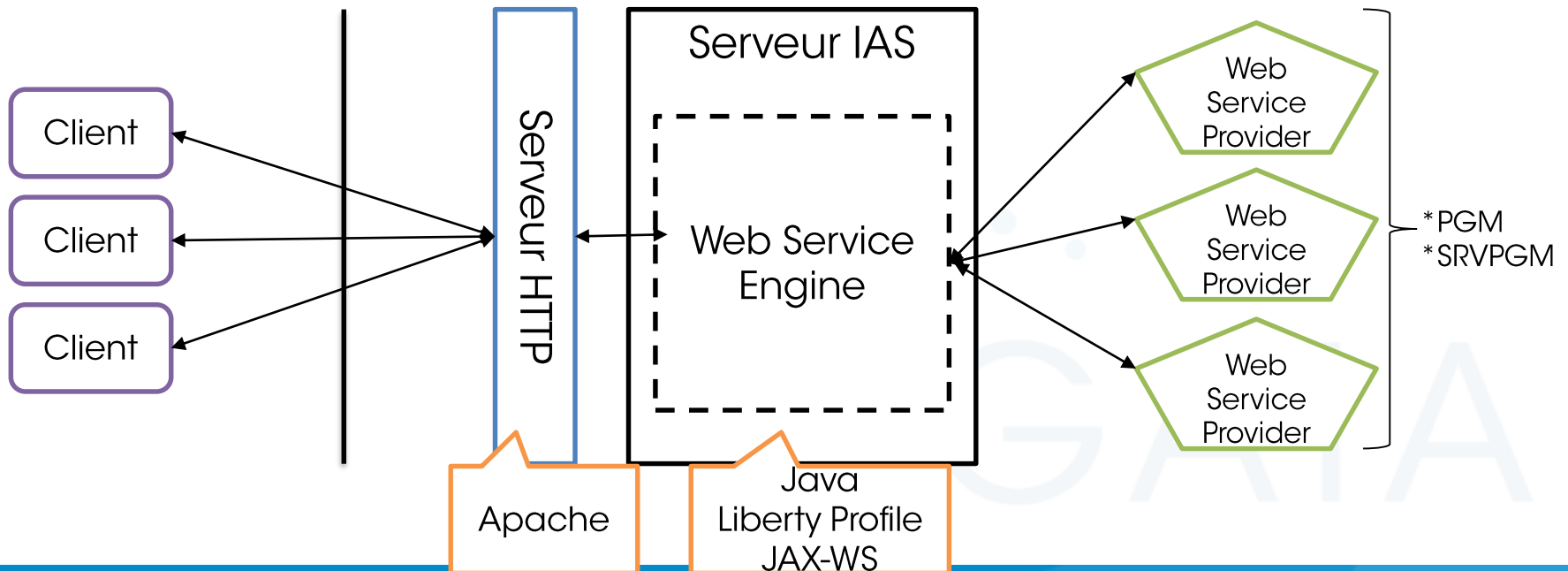
Fonctionnement IWS client et serveur

- Prérequis

- Extended Base Directory Support - base option 3 of operating system
- Host Servers - base option 12 of operating system
- System Openness Includes – base option 13 of operating system (only needed for web services client)
- Qshell - base option 30 of operating system
- PASE - base option 33 of operating system
- Digital Certificate Manager - base option 34 of operating system
- IBM HTTP Server for IBM i
- IBM Technology for Java SE 6 32 bit (i 7.1 and i 7.2), Java SE 7 32 bit (i 7.3) (only needed for web services server)

Fonctionnement IWS serveur

- But
 - Automatiser l'exposition de web services
 - depuis des exécutables ILE existants : Programmes, Programmes de service
- Schéma de principe



Fonctionnement IWS serveur

- Objectif
 - Permettre d'exposer des programmes en tant que web services en rendant la technologie transparente
- Moyens
 - Une application Java serveur dédiée
 - Capable de générer du code Java à la volée pour exposer un nouveau programme en tant que service web
 - Se base sur le toolbox Java
 - Une interface web
 - Permet de créer des serveurs, déployer des services, arrêter/démarrer serveurs et services
 - Des scripts shell



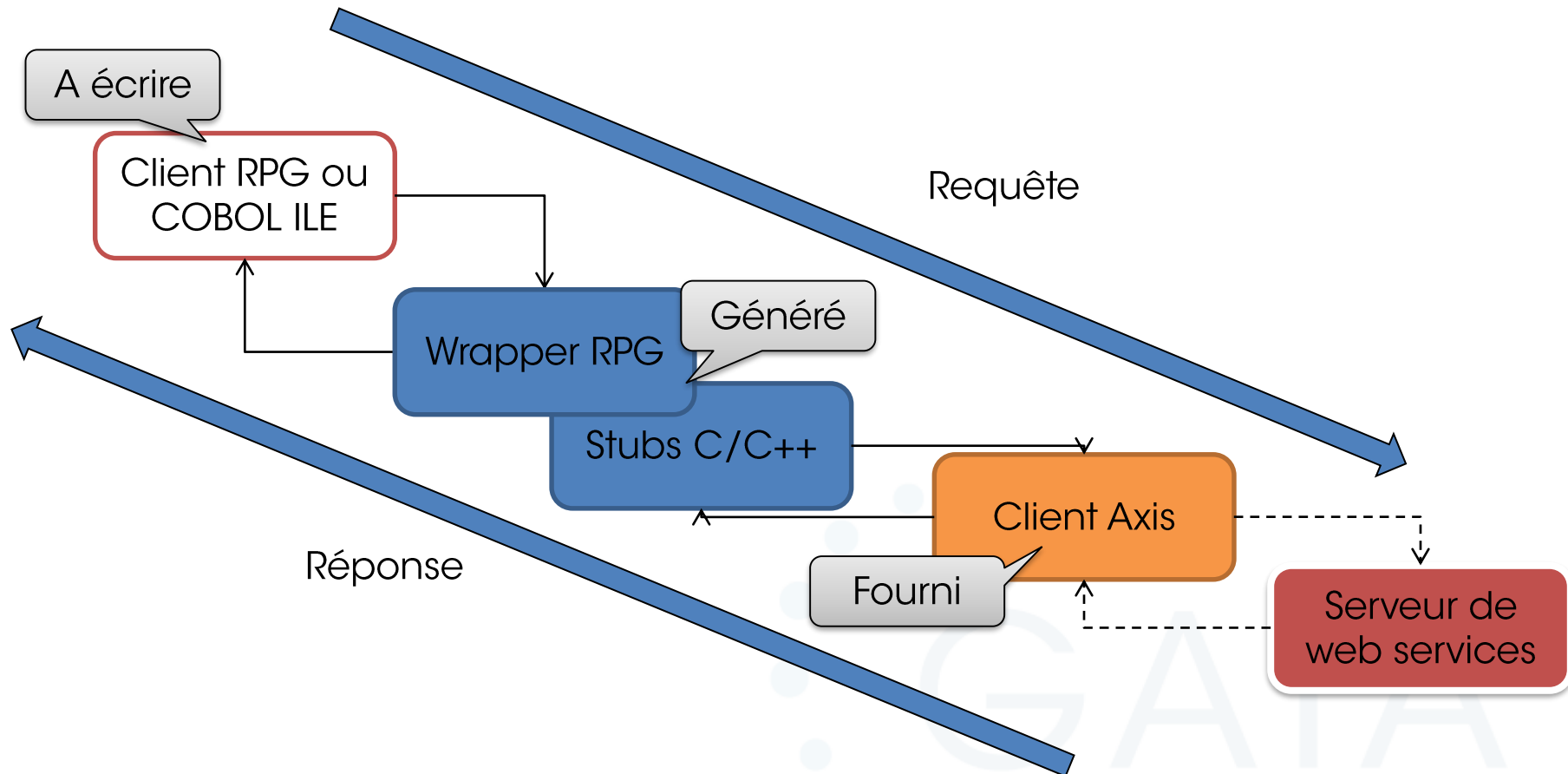
Fonctionnement IWS client

- But
 - Fournir les outils permettant l'appel d'un web service depuis un programme ILE
 - Possibilité d'appeler les services
 - REST
 - SOAP : dans ce cas, génération de code depuis le WSDL
 - Le client est entièrement « natif »
 - Du code compilé dans QSYS sous forme ILE
 - *MODULE /*PGM /*SRVPGM
 - Des « bibliothèques » dans l'IFS
 - Axis (projet Open Source Apache)



Fonctionnement IWS client

- Schéma de principe



Démo (10 min)



GAIA

Démo

- Déploiement service
 - REST
 - SOAP
- Appel service
 - REST
 - SOAP



Possibilités et limites



Possibilités et limites

- Côté client

- Langages clients supportés
 - Langages ILE : RPG, C, C++, COBOL
- Réseau
 - Support https, proxy
- SOAP
 - WS-I Basic Profile : WSDL 1.1 / SOAP 1.1
 - Restrictions sur le WSDL
 - Les dates échangées doivent être postérieures au 1er janvier 1970
 - Les pièces jointes ne sont pas supportées
 - Seul le style d'encodage « literal » est supporté
 - Encodage UTF-8
 - Un unique service avec un unique port par WSDL
 - ref, xsd:union et xsd:any non supportés
 - Gestion des exceptions soap:fault
 - faultcode, faultstring et faultactor remontent directement dans le code RPG
 - détail qui permet de personnaliser les erreurs ne remontent pas
 - » Possibilité de personnaliser la gestion des erreurs pour le récupérer
 - » Non documenté

Possibilités et limites

- Côté Serveur

- Possibilité pour le programme RPG
 - D'accéder aux métadonnées de la requête
 - De contrôler l'entête HTTP de la réponse
- La bibliothèque du programme n'est pas modifiable, pas de *LIBL
 - Le code Java généré contient la bibliothèque, compilé en dur !
- Pas de modification de l'utilisateur d'exécution du serveur
- Protocoles
 - WSDL 1.1 uniquement
 - SOAP 1.1 et 1.2
 - WS-Security et WS-Policy non supportés
- Paramètres
 - 32 pour un programme
 - 7 pour une procédure
- Limitation du PCML
 - Pointeur, entier signé sur 1 octet, entier non signé sur 8 octets
 - Valeurs de retour de procédure doit être un entier signé sur 4 octets
- Limitation RPG
 - 1 paramètre ne peut dépasser 16 Mo

Possibilités et limites

- JSON

- Aujourd'hui on utilise YAJL
- Ou tout autre solution pour lire/produire du JSON
 - Souvent manuellement également
- L'annonce de fonctionnalités JSON via DB2 va remédier à ce problème



Evolution



Support architecture 2-tiers

- Disponible avec les groupes HTTP suivants
 - V7R1M0 SF99368-Level 50
 - V7R2M0 SF99713-Level 24
 - V7R3M0 SF99722-Level 11
- Permet d'appeler un programme (de service) ILE sur un autre système
 - Paramétrable via l'interface web et les scripts shell

```
> setWebServiceProperties.sh -help
```

```
Command usage:
```

```
setWebServiceProperties.sh
```

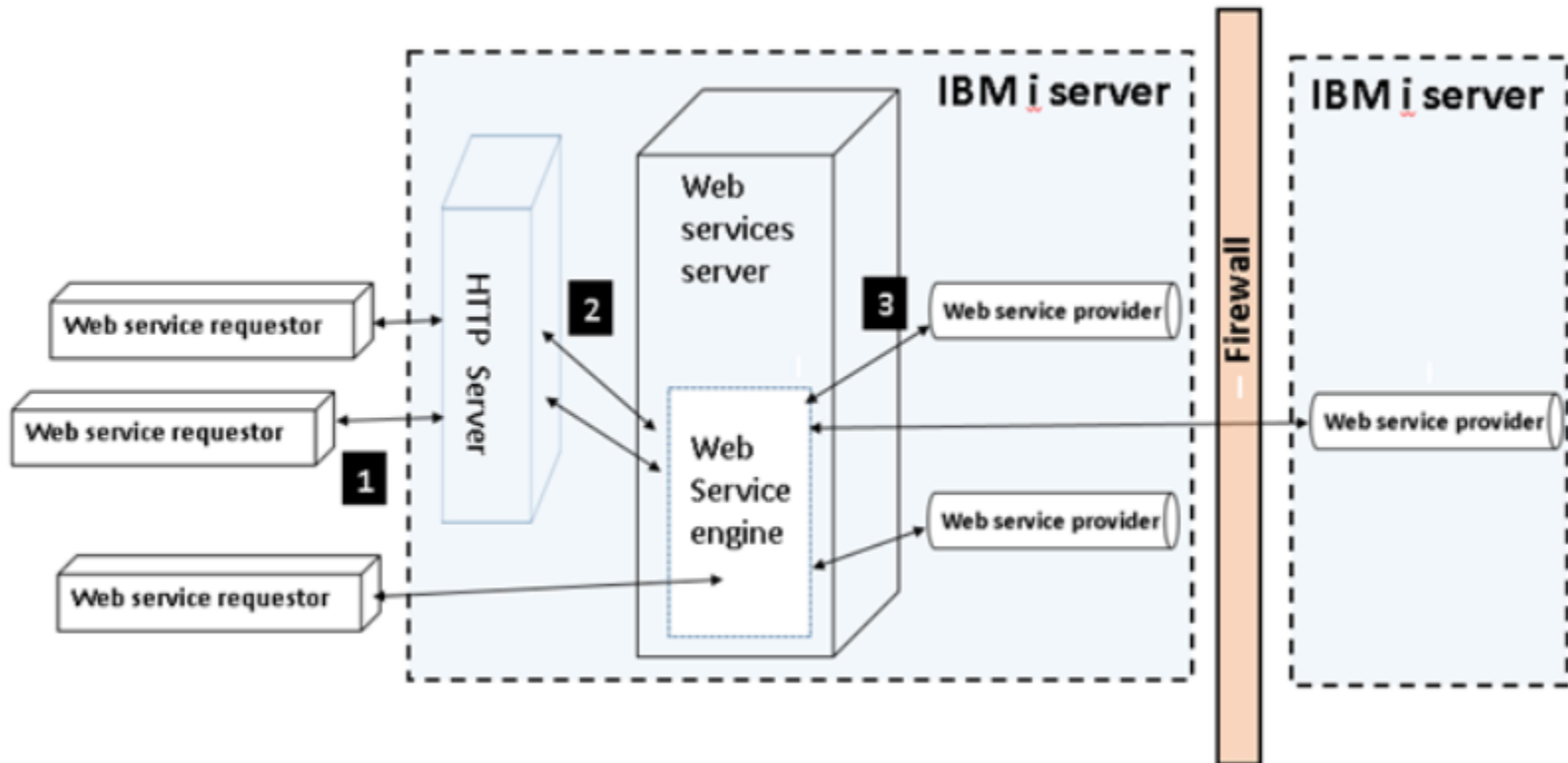
```
-server 'server-name' -service 'service-name'
```

```
[ -userid 'userid' ] [ -host 'host-server' ] [ -resetWSDL ]
```

```
[ -libraryList 'library-list' ] [ -libraryListPosition '*FIRST|*LAST' ]
```

Support architecture 2-tiers

- Schéma de principe



<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/dW%20IBM%20Integrated%20Web%20Services%20for%20i/page/Support%20for%202-tier%20web%20services>

Nouvelles options de l'interface

- Serveur
 - Gestion des certificats SSL pour connexion sécurisée entre les IBM i

Properties

Display and manage the properties of the application server.

Application Server Ports JVM Options Web Services

Property information for the integrated Web application server ?

Version: 8.5
Subsystem: QHTTPSVR
Job name:
User ID: NB
Instance path: /www/ws_test/wlp/usr/servers/ws_test
JAVA home: /QOpenSys/QIBM/ProdData/JavaVM/jdk71/32bit ▼
Default key store: *SYSTEM ▼
Key store password:

Nouvelles options de l'interface

- Service
 - Précision de l'hôte sur lequel exécuter la requête

Service Properties

General Methods Library List Swagger Connection Pool Request Information

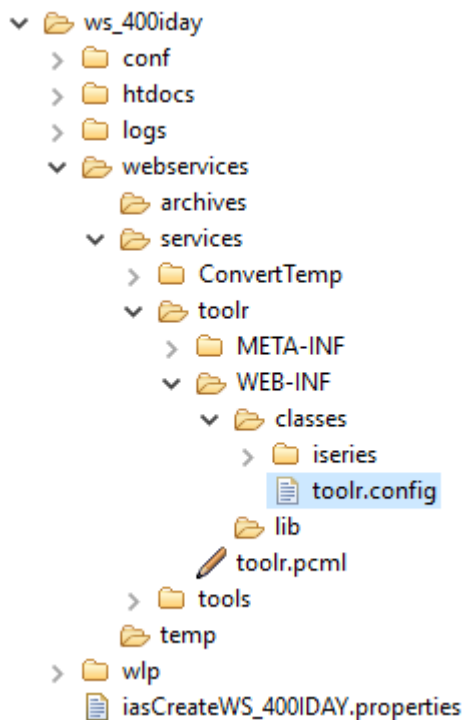
Connection pool information ?

Host server:	localhost	
Use secure connections:	No ▼	
Maximum number of connections:	*NOMAX	or... ▼
Maximum connection use count:	*NOMAX	or... ▼
Maximum connection use time (seconds):	*NOMAX	or... ▼

: GAIA

Configuration

- La configuration du service
 - Pas d'interface (encore) pour le mot de passe



```
*toolsr.config
#Configuration File - QIWS
#Wed Oct 25 10:51:29 CEST 2017
WDT_IASPNAME=
WDT_CONNPOOLCCSID=-1
WDT_PASSWORDENCODING_ENABLE=false
WDT_HOSTNAME=as400
WDT_RUNTIMELIB=OUTILS;
WDT_EXCEPTIONONERROR=true
WDT_CONNPOOLUSETHREADS=true
WDT_CONNPOOLCLEANUPINTERVAL=300000
WDT_TRANSPORTMETADATA=
WDT_RUNTIMECURLIB=
WDT_TRANSPORTHEADERS=
WDT_USESSL=false
WDT_CONNPOOLMAXLIFETIME=86400000
WDT_TRACEFILE=
WDT_CONNPOOLMAXUSETIME=-1
WDT_RUNTIMELIBIFS=OUTILS;
WDT_PASSWORD=xxxxxxxxx
WDT_USERID=NB
WDT_PGMOBJECTPATH=/QSYS.LIB/IDAY4.LIB/WSTOOLS.SRVPGM
WDT_GENERATEDWSDLTIME=1508834362000
WDT_RUNTIMELIBPOSITIONS=*LAST;
WDT_RUNTIMEINITCMD=
WDT_CONNPOOLMAXINACTIVITY=3600000
WDT_TRACE=
```

Serveur Administration et exploitation



Administration et Exploitation des serveurs IWS

- Interface web adaptée au développement, test,...
 - Pas à l'exploitation / administration
- Tâches courantes d'exploit / admin
 - Arrêt / redémarrage
 - Services
 - Serveurs
 - Pour gestion sauvegarde, redémarrage après IPL, gestion des environnements...
 - Déploiement
 - De développement vers systèmes de recette, production
 - Surveillance
 - Service en fonctionnement
 - Exceptions



Scripts

- **Scripts shell**

- Permettent d'automatiser, donc d'outiller
- Disponibles dans /QIBM/ProdData/OS/WebServices/bin/

Nom	Suite
createWebServicesServer.sh	saveWebServices.sh
deleteWebServicesServer.sh	saveWebServicesServer.sh
getWebServiceProperties.sh	setWebServiceProperties.sh
getWebServicesServerProperties.sh	setWebServicesServerProperties.sh
installWebService.sh	startWebService.sh
listWebServices.sh	startWebServicesServer.sh
listWebServicesServers.sh	stopWebService.sh
restoreWebServices.sh	stopWebServicesServer.sh
restoreWebServicesServer.sh	uninstallWebService.sh

Arrêt / démarrage

- Arrêt d'un service

- `stopWebService.sh -server 'ws_400iday' -service 'convertTemp'`

- Arrêt d'un serveur

- `stopWebServicesServer.sh -server 'ws_400iday'`

- Fonctionnement

- Les commandes qsh sont asynchrones
- Renvoient un code
 - IWS00102I - Command completed successfully.
 - IWS00206E - Web service failed to stop.
 - IWS00240E - Server does not exist.
- Possibilité d'utilisation en CL
 - `QSH CMD('cd /QIBM/ProdData/OS/WebServices/bin/; stopWebService.sh -server 'ws_400iday' -service 'convertTemp''')`

Déploiement

- Le déploiement peut se faire via l'interface web ou par script
 - Certaines options ne sont présentes que dans l'une ou l'autre des solutions... à la marge
 - Les scripts permettent aussi un déploiement par
 - Sauvegarde
 - Restauration sur une autre instance, sur le même système ou non



Déploiement

- SOAP

```
installWebService.sh
  -server 'ws_400iday'
  -programObject '/QSYS.LIB/IDAY4.LIB/WSTOOLS.SRVPGM'
  -service 'tools'
  -userid 'NB'
  -detectFieldLengths
  -serviceType '*SOAP11'
  -parameterUsage 'ECHO:i,o,o:JOBLOG:i,o,o'
  -libraryList 'NB;OUTILS'
  -libraryListPosition '*LAST'
  -printErrorDetails
```



Déploiement

- REST

- Possible via le script, mais nécessite en général un fichier de propriétés supplémentaires pour donner du détail
- Fichier `wstools.properties`

```
uri.path.template=/wstoolsrest
```

```
ECHO.uri.path.template=/echo/{str}  
ECHO.wrap.input.parameters=false  
ECHO.wrap.output.parameter=true  
ECHO.http.request.method=GET  
ECHO.consumes=/*/*  
ECHO.produces=application/xml, application/json  
ECHO.response.code.parameter=  
ECHO.http.headers.parameter=  
ECHO.strIn.usage=input  
ECHO.strIn.pathparam=str  
ECHO.strOut.usage=output  
ECHO.dateOut.usage=output
```

```
JOBLOG.uri.path.template=/joblog/{cd}  
JOBLOG.wrap.input.parameters=false  
JOBLOG.wrap.output.parameter=true  
JOBLOG.http.request.method=GET  
JOBLOG.consumes=/*/*  
JOBLOG.produces=application/xml, application/json  
JOBLOG.response.code.parameter=  
JOBLOG.http.headers.parameter=  
JOBLOG.code.usage=input  
JOBLOG.code.pathparam=cd  
JOBLOG.error.usage=output  
JOBLOG.dateOut.usage=output
```

Déploiement

- REST (suite)

```
installWebService.sh
```

```
-server 'ws_400iday'
```

```
-programObject '/QSYS.LIB/IDAY4.LIB/WSTOOLS.SRVPGM'
```

```
-service 'toolr'
```

```
-userid 'NB'
```

```
-detectFieldLengths
```

```
-serviceType '*REST'
```

```
-propertiesFile '/home/iday/201711/wstools.properties'
```

```
-libraryList 'OUTILS'
```

```
-libraryListPosition '*LAST'
```

```
-printErrorDetails
```



Déploiement







- Résultat

[ws_400iday](#) > Manage Deployed Services

Manage Deployed Services

Data current as of 24 oct. 2017 10:39:27.

Deployed services: [?](#)

	Service name	Status	Type	Startup type	Service definition
<input type="radio"/>	ConvertTemp	 Running	SOAP	Automatic	 View WSDL
<input type="radio"/>	toolr	 Running	REST	Automatic	 View Swagger
<input checked="" type="radio"/>	tools	 Running	SOAP	Automatic	 View WSDL

Deploy

Stop

Properties

Uninstall

Refresh

G A I A

Personnalisation

- Serveur

- Certaines caractéristiques du serveur sont personnalisables
 - JVM et options de la JVM
 - Adresse IP et port du serveur HTTP et Java
 - Le « context root », c'est-à-dire la base de l'URL d'appel des services

Properties

Application Server Ports JVM Options **Web Services**

Property information for the integrated Web services server ?

Version: Apache CXF 2.6
Services path: /www/ws_400iday/webservices/services
Context root name:


Cf fichier /www/*instance*/wlp/usr/servers/*instance*/bootstrap.properties

Personnalisation

- Services

- Pour tous les services, possibilités de modifier principalement, par script, interface web ou fichiers de conf
 - Démarrage automatique avec le serveur ou non
 - Utilisateur pour l'exécution
 - Liste de bibliothèques
 - Paramètres du pool de connexions

Service Properties

General	Operations	Library List	WSDL	Connection Pool	Request Information
Connection pool information 					
Maximum number of connections:	<input type="text" value="*NOMAX"/>	<input type="text" value="or..."/>	<input type="text" value="▼"/>		
Maximum connection use count:	<input type="text" value="*NOMAX"/>	<input type="text" value="or..."/>	<input type="text" value="▼"/>		
Maximum connection use time (seconds):	<input type="text" value="*NOMAX"/>	<input type="text" value="or..."/>	<input type="text" value="▼"/>		
Maximum connection inactivity (seconds):	<input type="text" value="3600"/>	<input type="text" value="or..."/>	<input type="text" value="▼"/>		
Maximum lifetime (seconds):	<input type="text" value="86400"/>	<input type="text" value="or..."/>	<input type="text" value="▼"/>		
Connection CCSID:	<input type="text" value="*USERID"/>	<input type="text" value="or..."/>	<input type="text" value="▼"/>		
Cleanup interval (seconds):	<input type="text" value="300"/>				
Use maintenance threads:	<input type="text" value="Yes"/>				



Personnalisation

- Services SOAP

- Personnalisation du WSDL
- Celui généré n'est pas très « précis »
- Exemple

- Message entrée

```
<xs:complexType name="jobLogInput">  
  <xs:sequence>  
    <xs:element name="code" type="xs:string"></xs:element>  
  </xs:sequence>  
</xs:complexType>
```

- Message sortie

```
<xs:complexType name="jobLogResult">  
  <xs:sequence>  
    <xs:element name="error" type="xs:string"></xs:element>  
    <xs:element name="dateOut" type="xs:date"></xs:element>  
  </xs:sequence>  
</xs:complexType>
```

Personnalisation

- Problème

- En XML, un string a une taille en 0 et * caractères
 - Or, le serveur plante lorsqu'on lui transmet une valeur dont la taille est supérieure à la taille du paramètre !

- Usage


- Si vous transmettez les WSDL à des partenaires ou clients, ils doivent être détaillés au maximum



Personnalisation

- Exemple modification
 - Valeur obligatoire, définition de la longueur, domaine de valeur

```
<xs:complexType name="jobLogInput">
  <xs:sequence>
    <xs:element name="code" minOccurs="1" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>Code traitement (0 = Joblog, N = pas de joblog)</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xsd:string">
          <xs:enumeration value="0"></xs:enumeration>
          <xs:enumeration value="N"></xs:enumeration>
          <xs:minLength value="0"></xs:minLength>
          <xs:maxLength value="1"></xs:maxLength>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The logo for GAIA is located in the bottom right corner of the slide. It consists of the word "GAIA" in a large, light blue, sans-serif font. To the left of the text are five small, light blue circles of varying sizes, arranged in a pattern that suggests a globe or a network.

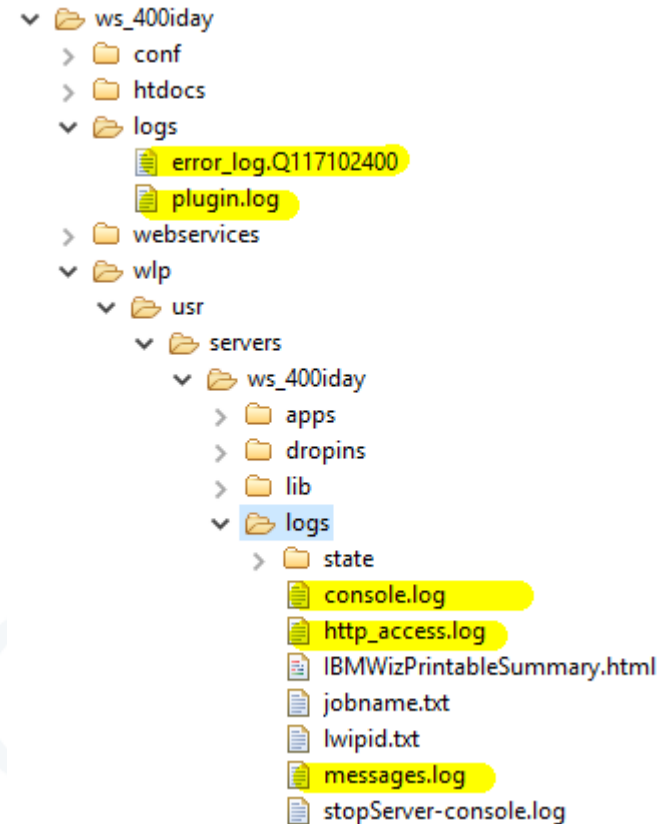
Personnalisation

- Services REST
 - Personnalisation du SWAGGER possible
 - Toutefois, celui-ci est mieux construit

```
"paths": {
  "/wstoolsrest/echo/{str}": {
    "get": {
      "tags": [
        "tools APIs"
      ],
      "operationId": "echo",
      "consumes": [
        "*/*"
      ],
      "produces": [
        "application/xml",
        "application/json"
      ],
      "parameters": [
        {
          "name": "str",
          "in": "path",
          "required": true,
          "type": "string",
          "maxLength": 25
        }
      ]
    }
  },
}
```

Surveillance

- Présence du service
 - Les jobs doivent être présents : surveillance classique
- Bon fonctionnement du service
 - Le serveurs génèrent des logs
 - dans l'IFS



Surveillance

- Vous avez probablement des outils capables de scruter les logs
 - Largement utilisés dans le monde Unix/Linux et Windows
 - Créer un partage NetServer pour accéder aux logs et scruter les messages d'erreur
- Vous pouvez aussi utiliser la fonction « Web Log Monitor »
 - Sur chaque instance de serveur
 - Scrute un fichier de log par rapport à une règle
 - Mot ou expression régulière
 - Déclenche une notification dans *SYSOPR / mail
 - Souvent *SYSOPR est surveillée, ou permet également de déclencher une action automatiquement

Surveillance

- Exemple

File and Rule Notification Miscellaneous

File and rule configuration:

Log File	Configuration				
<input type="checkbox"/> /www/ws_400iday/wlp/usr/servers/ws_400iday/logs/console.log	Associated rule and notification text :				
	<table border="1"><thead><tr><th>Rule</th><th>Notification Text</th></tr></thead><tbody><tr><td>[Include]"erreur"</td><td>File /www/ws_400iday/wlp/usr/servers/ws_400iday/logs/console.log has something...</td></tr></tbody></table>	Rule	Notification Text	[Include]"erreur"	File /www/ws_400iday/wlp/usr/servers/ws_400iday/logs/console.log has something...
Rule	Notification Text				
[Include]"erreur"	File /www/ws_400iday/wlp/usr/servers/ws_400iday/logs/console.log has something...				

Add Modify Remove

File and Rule Notification Miscellaneous

Notification:

Send message to *QSYSOPR message queue.

Send email to e.g. joe@us.i

Sender's email address:

SMTP server:

Account:

Password:

File and Rule Notification Miscellaneous

Monitor interval:

Check the log file every: Minutes ▾

Notification flood prevention:

Maximum permitted number of notifications per hour:

Surveillance

- Exemple

/www/ws_400iday/wlp/usr/servers/ws_400iday/logs/console.log

Lancement de ws_400iday (WebSphere Application Server 17.0.0.2/wlp-1.0.17.cl170220170523-1818) sur IBM J9 VM, version jvmap3280sr4fp10-20170727_012.

```
[AUDIT ] CWNKE0001I: Le serveur ws_400iday a démarré.
[AUDIT ] CWNKT0016I: Application Web disponible, (default_host) : http://gaia:10022/web/services/toolr/
[AUDIT ] CWNKZ0001I: Application toolr démarrée en 0,964 secondes.
[AUDIT ] CWNKT0016I: Application Web disponible, (default_host) : http://gaia:10022/web/services/toolsService/
[AUDIT ] CWNKZ0001I: Application tools démarrée en 1,732 secondes.
[AUDIT ] CWNKT0016I: Application Web disponible, (default_host) : http://gaia:10022/web/services/ConvertTempService/
[AUDIT ] CWNKZ0001I: Application ConvertTemp démarrée en 1,741 secondes.
[AUDIT ] CWNKF0012I: Le serveur a installé les fonctions suivantes : [servlet-3.1, jaxb-2.2, ssl-1.0, jndi-1.0, jaxws-2.2, json-1.0, localConnec
[AUDIT ] CWNKF0011I: Le serveur ws_400iday est prêt pour une plateforme plus intelligente.
```

```
[AVERTISSEMENT] Application {http://tools.wsbeans.iseries/}tools#{http://tools.wsbeans.iseries/}echo has thrown exception, unwinding now
com.ibm.as400.access.ExtendedIllegalArgumentException: javaValue (plus long que 25 caractères c est trop pour le serveur): Length is not valid.
```

```
[24/10/17 11:29:04:702 CEST] 00000043 com.ibm.ws.session.WsSessionImpl
[24/10/17 11:18:04:404 CEST] 00000043 com.ibm.ws.util
[24/10/17 11:28:16:957 CEST] 00000086 com.ibm.ws.webcontainer.servlet
[24/10/17 11:29:13:083 CEST] 000000a1 org.apache.cxf.phase.PhaseInterceptorChain
org.apache.cxf.interceptor.Fault: com.ibm.as400.access.ExtendedIllegalArgumentException: javaValue (plus long que 25 caractères c est trop pour le serveur): Length is not valid.
at org.apache.cxf.service.invoker.AbstractInvoker.createFault(AbstractInvoker.java:162)
at org.apache.cxf.jaxws.AbstractJAXWSMethodInvoker.createFault(AbstractJAXWSMethodInvoker.java:213)
at org.apache.cxf.service.invoker.AbstractInvoker.invoke(AbstractInvoker.java:128)
at org.apache.cxf.jaxws.AbstractJAXWSMethodInvoker.invoke(AbstractJAXWSMethodInvoker.java:178)
at org.apache.cxf.jaxws.JAXWSMethodInvoker.invoke(JAXWSMethodInvoker.java:66)
at org.apache.cxf.service.invoker.AbstractInvoker.invoke(AbstractInvoker.java:75)
at org.apache.cxf.interceptor.ServiceInvokerInterceptor$1.run(ServiceInvokerInterceptor.java:58)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:522)
at java.util.concurrent.FutureTask.run(FutureTask.java:277)
at org.apache.cxf.workqueue.SynchronousExecutor.execute(SynchronousExecutor.java:37)
at org.apache.cxf.interceptor.ServiceInvokerInterceptor.handleMessage(ServiceInvokerInterceptor.java:107)
at org.apache.cxf.phase.PhaseInterceptorChain.doIntercept(PhaseInterceptorChain.java:262)
at org.apache.cxf.transport.ChainInitiationObserver.onMessage(ChainInitiationObserver.java:121)
at org.apache.cxf.transport.http.AbstractHTTPDestination.invoke(AbstractHTTPDestination.java:211)
at com.ibm.ws.jaxws.endpoint.AbstractJaxWsWebEndpoint.invoke(AbstractJaxWsWebEndpoint.java:182)
at com.ibm.ws.jaxws.webcontainer.LibertyJaxWsServlet.handleRequest(LibertyJaxWsServlet.java:135)
at com.ibm.ws.jaxws.webcontainer.LibertyJaxWsServlet.doPost(LibertyJaxWsServlet.java:84)
```

Sécurité



Sécurité

- Serveur
 - Utilisation de certificat SSL
 - Créé via une autorité locale
 - Importé via une autorité de confiance
 - Nominatif (permettant l'authentification)
 - Utilisation d'une authentification HTTP
 - http basic authentication
 - Soit par rapport à un profil IBM, une liste de validation, un « profil web »
 - Sécurisation complémentaire
 - Via vos équipements réseau (firewall, reverse proxy ...)
 - Via des solutions logicielles : jetons, clés, authentification métier ...
 - Conseil
 - Générer des logs !

Sécurité

- Client

- Capacité de se connecter en https
 - Gestion des certificats via DCM ou autre magasin de certificats
 - Commande keytool
- Support http basic authentication
- Support de la connexion au proxy



Sécurité

- WS Security

- Cette extension de la norme WS est la plus fréquemment utilisée
 - https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
 - Mise en place
 - Signature et chiffrement du message SOAP
 - Jeton de sécurité pour identifier l'émetteur
- Non supporté sur le serveur et le client



DB2

SQL en tant que client ?

- Résumé des fonctions disponibles dans DB2
 - XML natif
 - Stockage, publication et décomposition
 - JSON
 - Partiel aujourd'hui : stockage et décomposition
 - Publication à venir
 - HTTP
 - GET, POST, PUT, DELETE
 - Avec gestion des entêtes
 - Utilitaires divers
 - Encodage/décodage URL, base64, ccsid ...
- Donc tout ce qui est nécessaire à l'appel d'un service web
 - REST
 - XML / JSON
 - SOAP
 - Message XML à construire et décomposer manuellement

GAIA

Exemple REST

- Echo

```
SELECT x.strout , systools.urldecode(x.strout, 'UTF-8')
FROM
  json_table(systools.httpgetclob(
    'http://neptune:10032/web/services/toolr/wstoolsrest/echo/'
    concat systools.urlencode('Valeur ici', NULL),
    '<httpHeader><header name="Accept" value="application/json"
/></httpHeader>')
, 'lax $' columns(strOut VARCHAR(25) PATH 'lax $.strOut'))
AS X;
```

STROUT	00002
Valeur+ici	Valeur ici

Exemple SOAP

- Echo

```
SELECT x.strout , x.dateOut
      FROM xmltable('/*:Envelope/*:Body/*:echoResponse'
                    passing XMLPARSE(document
systools.httppostclob('http://neptune:10032/web/services/toolsService/tools'
, '' ,
                        XMLDOCUMENT( XMLELEMENT(NAME "soapenv:Envelope",
XMLNAMESPACES('http://schemas.xmlsoap.org/soap/envelope/' AS "soapenv",
'http://tools.wsbeans.iseries/' AS "tool") ,
                        XMLELEMENT(NAME "soapenv:Body",
                                XMLELEMENT(NAME "tool:echo",
                                        XMLELEMENT(NAME "arg0",
                                                XMLELEMENT(NAME "strIn" , 'Valeur
accentuée'))))))))
      columns strOut CHAR(25) PATH 'return/strOut',
             dateOut DATE PATH 'fn:adjust-date-to-
timezone(return/dateOut,())'
      AS X;
```

STROUT	DATEOUT
Valeur accentuée	2017-10-24

Comparatif

	DB2	ILE natif
Compétences requises	http, XML, JSON, DB2	http, XML, JSON, ILE
Facilité / quantité de code	Moins de code	Plus de code
Technologie	Java	ILE + Axis
Fonctionnalités REST/SOAP	Tout est faisable manuellement	Tout est faisable avec customisation Automatisme pour SOAP
Réseau, proxy	Via conf Java	Via code
Contrôle environnement exécution	Non	Oui Capacité à faire des traces, handler...
Performances	Bonnes Instanciation JVM 1 ^{er} appel	Bonnes
SSL	Oui via magasins de certificats de la JVM (import par keytool)	Oui via DCM où autre magasin de certificats

Retour d'expérience

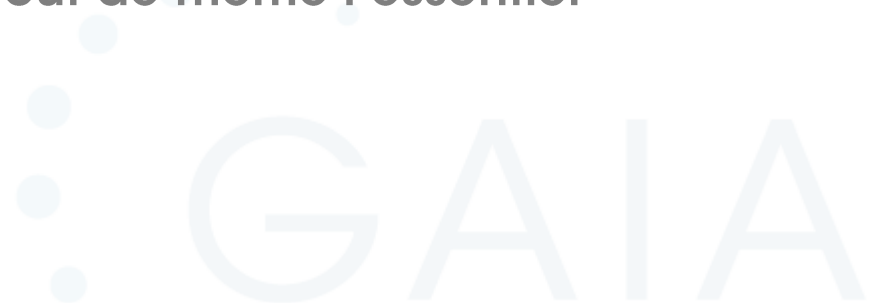
Nos projets clients, retour d'expérience

- Projets client
 - Typologie de projet
- Sur 31 clients ou prospects sondés
 - 9 possèdent déjà des web services
 - Dont 3 avec plus de 10 web services
 - 12 ont des projets
 - Dont des projets avancés pour 3 d'entre eux
 - Sur les 10 qui restent
 - 5 ont IBM i « Stabilisés »
 - 5 autres ne savent pas



Nos projets clients, retour d'expérience

- Retour d'expérience global
 - Difficulté sur les dates / heures / horodatage
 - Format C/Linux pour le client
 - Support PCML tardif pour le serveur
 - Gestion du timezone
 - Difficulté pour les équipes
 - Multiplicité des technos à manipuler
 - Pas de problème technologique
 - **Cela fonctionne ! Et c'est tout de même l'essentiel**



Nos projets clients, retour d'expérience

- Retour d'expérience Client

- ILE

- Bonnes performances
 - Utilisation de DB2 pour la partie XML, souvent de YAJL pour la partie JSON (notre préconisation jusqu'à maintenant)
 - Capacité à faire des traces, personnaliser les traces : appréciable
 - Gestion des exceptions (soap:fault) automatisée
 - Sauf lorsque personnalisées : la récupération de l'exception par le code est à faire manuellement
 - SOAP : automatisation du code généré appréciable
 - 3 procédures à appeler pour coder un client

- DB2

- Peu de retour encore
 - La disponibilité de fonctions de publication JSON via DB2 devrait rendre l'option DB2 plus attractive
 - Pas de trace disponible, la gestion des erreurs en sera plus complexe

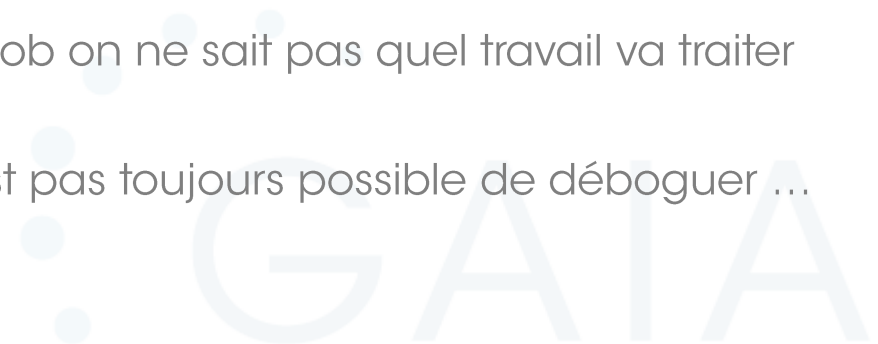
Nos projets clients, retour d'expérience

- Retour d'expérience Serveur
 - Bonnes performances
 - Comment s'organise-t-on ?
 - Plusieurs instances ?
 - Impact sur la gestion de l'exploitation
 - Plusieurs services ou un service et plusieurs opérations ?
 - Gestion de la maintenance
 - Programmation
 - 7 paramètres pour les procédures (sic !). Limitation déjà connue du toolbox Java
 - Status
 - Header
 - *_LENGTH
 - Impossible de gérer les erreurs de façon souple
 - SOAP : impossible de renvoyer des soap:fault personnalisées
 - REST : impossible de renvoyer un message différent, mais possibilité de personnaliser le code HTTP

Nos projets clients, retour d'expérience

○ Maintenance

- Bibliothèque en dur
 - Lors du déploiement, les classes Java générées comportent la bibliothèque du programme [de service] en dur !
- obligation de suppression + republication
 - Principalement gênant dans le cas programme de service (ie 1 service avec plusieurs opérations)
- Versionning des services lors des évolutions ?
 - Pas de recette miracle ici !
- Débogage
 - Dans le cas de pools de job on ne sait pas quel travail va traiter
 - RDi utile
 - Générer des traces, il n'est pas toujours possible de déboguer ...



Nos projets clients, retour d'expérience

○ Personnalisation

- Limitation dans ce que l'on peut modifier dans les URL
 - `http://whost>:<port>/<context-root>/<root-resource>/uri-path-template>`
 - » Principalement `context-root` est commun au serveur

○ Surveillance

- Peu pratique : si vous avez des outils de surveillance des logs
- Par de remonté de métrique de performance, de volume d'appel etc ... à outiller vous-même si nécessaire
- Logs peu parlantes → faites vos logs applicatives,



Nos projets clients, retour d'expérience

- Quelques usages particuliers

- Transfert d'image

- Les attachements ne sont pas supportés par le serveur
 - Transmission en base64 dans des paramètres du programme

- URL

- L'impossibilité de personnaliser l'ensemble de l'URL
 - Mise en place d'un reverse proxy avec réécriture d'URL
 - Cela rend transparent au client final l'implémentation réelle du service



La suite

- Encore de nombreuses fonctionnalités utiles
 - Client en SOAP 1.2
 - Authentification SAML
 - ...
- De nombreuses évolutions demandées par les clients et prévues par IBM
 - <https://www.ibm.com/developerworks/rfe/execute>
 - 9 RFE sur 27 (RFE validés)



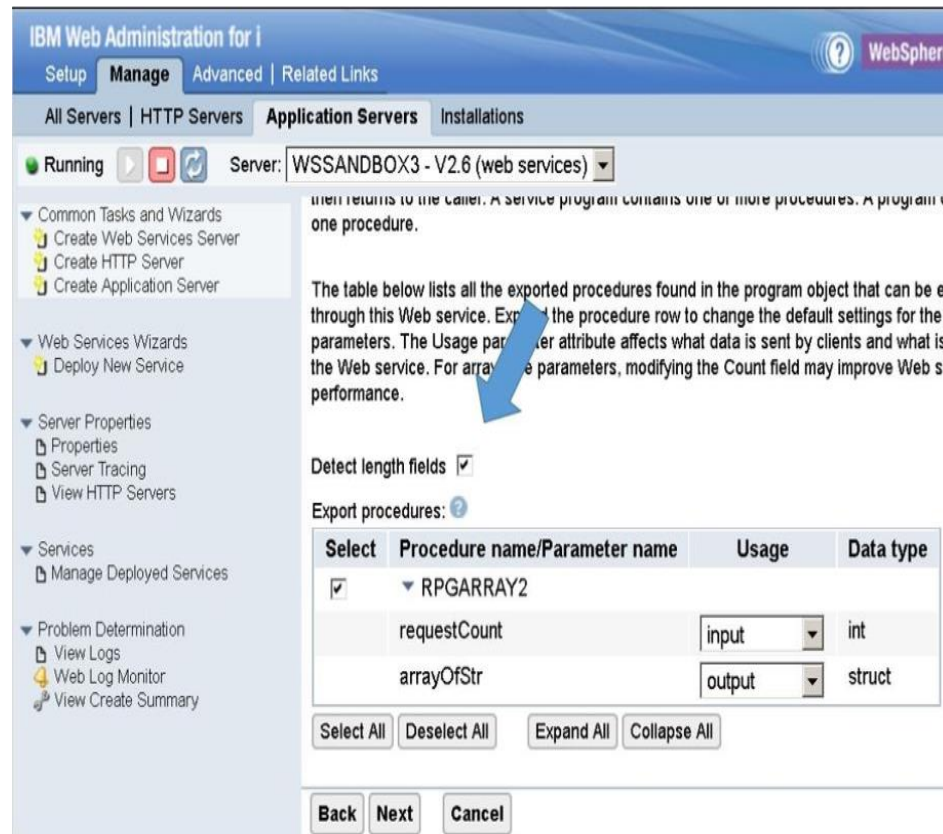
Console



Constat

- Aujourd'hui

- L'administration se fait de manière artisanale par le serveur WEB



The screenshot shows the IBM Web Administration console interface. The main content area displays the 'Export procedures' dialog for a web service. The server is identified as 'WSSANDBOX3 - V2.6 (web services)'. The dialog includes a table of exported procedures and a 'Detect length fields' checkbox.

Common Tasks and Wizards

- Create Web Services Server
- Create HTTP Server
- Create Application Server

Web Services Wizards

- Deploy New Service

Server Properties

- Properties
- Server Tracing
- View HTTP Servers

Services

- Manage Deployed Services

Problem Determination

- View Logs
- Web Log Monitor
- View Create Summary

Server: WSSANDBOX3 - V2.6 (web services)

The table below lists all the exported procedures found in the program object that can be exported through this Web service. Expand the procedure row to change the default settings for the parameters. The Usage parameter attribute affects what data is sent by clients and what is received by the Web service. For array parameters, modifying the Count field may improve Web service performance.

Detect length fields

Export procedures: ?

Select	Procedure name/Parameter name	Usage	Data type
<input checked="" type="checkbox"/>	▼ RPGARRAY2		
	requestCount	input	int
	arrayOfStr	output	struct

Select All Deselect All Expand All Collapse All

Back Next Cancel

Constat

- On est dans un mode unitaire proche du 'NEXT NEXT'
- En matière d'administration on a coutume de dire
 - A partir de 10 : on scripte
 - A partir de 100 : on outille
- Pour cela, on dispose de scripts shell
 - /QIBM/ProdData/OS/WebServices/bin/



Structure

- Un webservice

- est un ensemble de répertoires de configuration
- et il prend un status Running ou Stop

```
conf          DIR
htdocs        DIR
iasCreate (your webservices) STMF
logs          DIR
webservices   DIR
wlp           DIR
```

- Par défaut

- <Service_dir> = /www/(Your webservice)/webservices/services/(Your webservice)

Administration

- L'administration, c'est
 - Différentes actions
 - Création / Déploiement
 - Supervision
 - La mise en œuvre de la politique de sécurité
 - On ne parle pas de mise au point qui devra être fait par les équipes de développement



Administration

- On peut donc faire une console d'administration
 - Qui peut nous permettre de gérer
 - les serveurs
 - les services
 - Mais surtout de faire des opérations non prévues dans l'interface graphique
 - Sauvegarde et restauration des services
 - Duplication de services
 - De réaliser des commandes contrôle des statuts des serveurs et des services



Choix technologiques de notre « solution »

- Un interface 5250
 - Pour les administrateurs IBM i
 - Pas de serveur web a démarré
 - Plus proche du système (pour les SAVE, ...)
- Utilisation de XMLSERVICE pour traiter les infos SH
- Des commandes IBM i pour le trapping...



Une console pour manager servers et services

```
Fichier Edition Vue Communication Actions Fenêtre Aide
23/10/17 Management of servers
14:58:27

1 Start 2 Stop 3 Save 4 Display 8 Archives 9 Services

Server Status
-----
_PRCAPR Running
_PRCREA Running
_PRCREADEV Running
_9_PRCREAINTE Running
_PRCWDSL Stopped
_WSAPRDEV26 Running
_WSAPRFIC26 Running
_WSAPRINT26 Running
_WSAPRREA26 Running
```

```
I:Axyas03 - QPADEV0009
Fichier Edition Vue Communication Actions Fenêtre Aide
23/10/17 Management of webservices BERTHOIN
15:02:34 Server : PRCREAINTE

1 Start 2 Stop 3 Save 4 Display

Service Status
-----
_Collaborateur_ProfilsParMarque_V02 Running
_Contrat_CGAffaireRechercher_V01 Running
_Contrat_ResiliationAutoCertificatPreparer_V01 Running
_Contrat_ResiliationAutoControler_V01 Running
_Contrat_ResiliationAutoPositionner_V01 Running
_ConvertTemp Running
_Courtier_IndicateursRechercher_V01 Running
_Demande_ObtenerGestionnaireDedie_V01 Running
_Demande_ProduitAdherentProposer_V01 Running
_Demande_ReservationNumeroAdherent_V01 Running
_Demande_ReservationNumeroCourtier_V01 Running
_Edition_PositionnerARReclamation_V01 Running
_Exposition_BarreAvancementSuppression_V01 Running
_Exposition_ReceptionDemandeAffaire_V01 Running
```

Rappel du rôle voulu par notre console

- Une console c'est administrer l'existant
 - Arrêter
 - Redémarrer
 - Sauvegarder
- Mais aussi faire évoluer (à terme...)
 - Dupliquer un service
 - Le modifier
- Et encore gérer le déploiement...



Supervision

- Une commande IBM i qui renvoie l'état d'un service

- S'intègre dans CLLE

```
Retrieve satus off webservice (RTVSTSWEB)
Indiquez vos choix, puis appuyez sur ENTREE.
Name server . . . . . _____
Name Webservice . . . . . _____
Status . . . . . _____ Valeur alphanum
```

- S'intègre dans une sonde

```
DCL &STS *CHAR 10
RTVSTSWEB NAMESERVER('Controle_srl') +
NAMESERVIC('Annul_VAC_New') STATUS(&STS)
SNDUSRMSG MSG('LE SERVEUR EST :' *BCAT &STS) MSGTYPE(*INFO)
RTVSTSWEB NAMESERVER('PRCREAINT') +
NAMESERVIC('ConvertXemp') STATUS(&STS)
```

Rétro-ingénierie

- Génération d'un fichier par serveur interrogeable par SQL
 - SELECT SERVER, NAME, WTYPE, PGM_OBJECT, WPATH FROM DSPWEBSER

```
SELECT SERVER, NAME, WTYPE, PGM_OBJECT, WPATH FROM DSPWEBSER
```

SERVER	NAME	WTYPE	PGM_OBJECT	W
PRCREAINT	... Collaborateur_ProfilsParMarque_V02	... SOAP	/QSYS.LIB/PROREAPR.LIB/EFawe490.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... Contrat_CGAffaireRechercher_V01	... SOAP	/QSYS.LIB/I1744990R.LIB/EFawe447.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... Contrat_ResiliationAutoControler_V01	... SOAP	/QSYS.LIB/PROREAPR.LIB/EFawe382.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... Positionner_V01	... SOAP	/QSYS.LIB/PROREAPR.LIB/EFawe381.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... Rechercher_V01	... SOAP	/QSYS.LIB/QSYSDIR.LIB/QIWSSAMPLE.SRVPGM	... /www/PRCREAINT/webserv
PRCREAINT	... QuestionnaireDedie_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe418.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... AdherentProposer_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe440.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... NumeroAdherent_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe408.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... NumeroCourtier_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe417.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... Reclamation_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe380.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... DemandeAffaire_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe437.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... Reclamation_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe438.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... Pli_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe464.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_DeplacerProjVersAdhes_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe465.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_DocumentsActionsDemandeLister_V01	... SOAP	/QSYS.LIB/PROREAPR.LIB/EFawe471.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_DocumentsClePEGAAfficher_GEST_V02	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe455G.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_DocumentsClePEGAAfficher_V02	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe455.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_DocumentsClientAdherentAfficher_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe454.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_DocumentsProjetAfficher_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe467.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_EffacerVewer_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe459.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_IndexDocumentsSortantsLister_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe460.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_IndexerPli_V03	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe453.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_LibererPli_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe452.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_LotsCorbelleParMarque_V02	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe451.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_PiocherLot_V02	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe450.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_PliAjouterCommentaire_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe468.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_PliSupprimerCommentaire_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe469.PGM	... /www/PRCREAINT/webserv
PRCREAINT	... GED_PliSupprimerDocuments_V01	... SOAP	/QSYS.LIB/I1746500R.LIB/EFawe470.PGM	... /www/PRCREAINT/webserv

List service off server (DSPWEBS)

Indiquez vos choix, puis appuyez sur ENTREE.

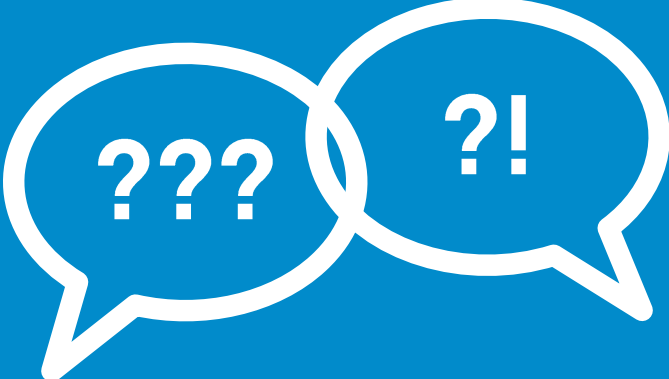
Name server

Rappel

- On a parfois besoin de savoir ce qui existe sans analyser l'élément par élément
- Par exemple
 - Pour connaître les programmes RPGLE qui ont été « webservisés »
 - Pour connaître la date de création du WS
 - ...



Vos projets ?



Références

- Liens

- https://fr.wikipedia.org/wiki/Service_web
- <https://fr.wikipedia.org/wiki/SOAP>
- <https://www.w3.org/TR/wsdl>
- <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- <https://www.w3.org/2001/sw/wiki/REST>
- <https://www.rfc-editor.org/rfc/rfc2068.txt>
- <https://www.w3.org/2005/Talks/1115-hh-k-ecows>

- IBM

- <https://www-03.ibm.com/systems/power/software/i/iws/>
- <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/dW%20IBM%20Integrated%20Web%20Services%20for%20i>
- https://www-03.ibm.com/systems/resources/systems_i_software_iws_pdf_WebServicesServer_new.pdf
- https://www-03.ibm.com/systems/resources/systems_i_software_iws_pdf_WebServicesClient_new.pdf
- <https://www.ibm.com/developerworks/ibmi/library/i-send-receive-user-defined-soap-rest-messages-trs/index.html>
- <https://www.ibm.com/developerworks/ibmi/library/i-integrated-web-services-server/index.html?ca=drs->
- <https://www.ibm.com/developerworks/ibmi/library/i-rest-web-services-server1/>
- <https://www.ibm.com/developerworks/ibmi/library/i-rest-web-services-server2/>
- <https://www.ibm.com/developerworks/ibmi/library/i-rest-web-services-server3/>