



# Graphical IPC UI with Adobe Flex®

Using Web Services and Default  
Values Management



**CWG**

Monday 20<sup>th</sup> 2009

# Configuration in Schneider-Electric

*A large range of products from simple to highly configurable*



*Simple products*



*Adapted products*



*Complex products*



# Configuration in Schneider-Electric

- **Over 1500 VC models, managed at different levels:**
  - Created by the central business units
  - Localized for/by each country and/or local business
  
- **Configuration models are integrated in sales and production processes**
  - **OPS** (Offer Presentation System): A custom designed information system that manages marketing and sales product data
  - **BRIDGE**: A central system for supporting the production process
  
- **A global organization with central and offshore resources**



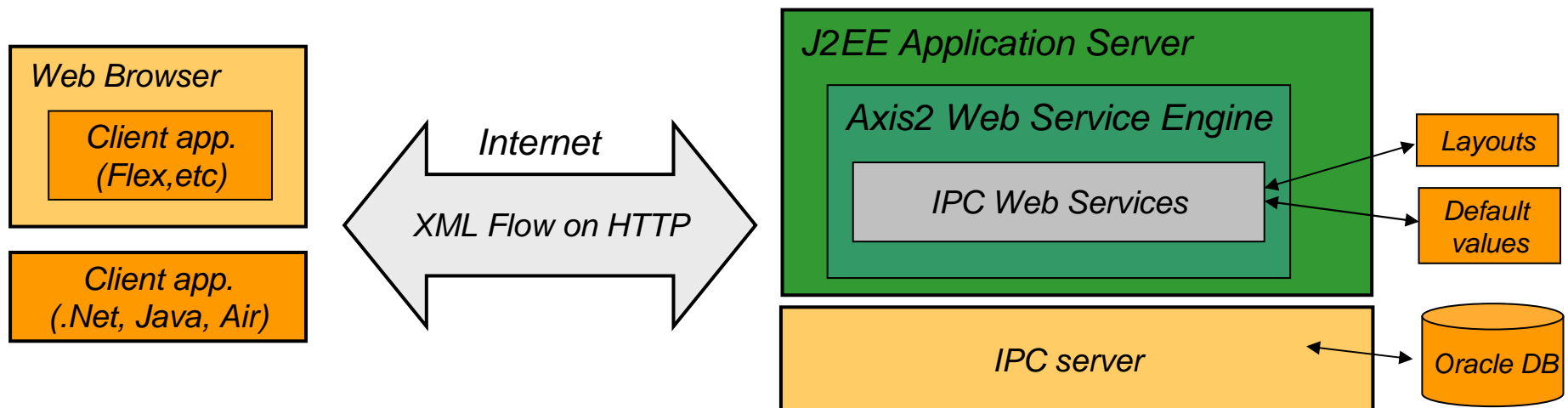
# Requirement: On-line/Off-line configurators

- Configuration engine for online applications, web catalogs, etc.
- Offline configuration engine for electrical network design and catalog software
  - Circuit design software is distributed to customers with embedded IPC
- Need a common interface for on-line and off-line configuration



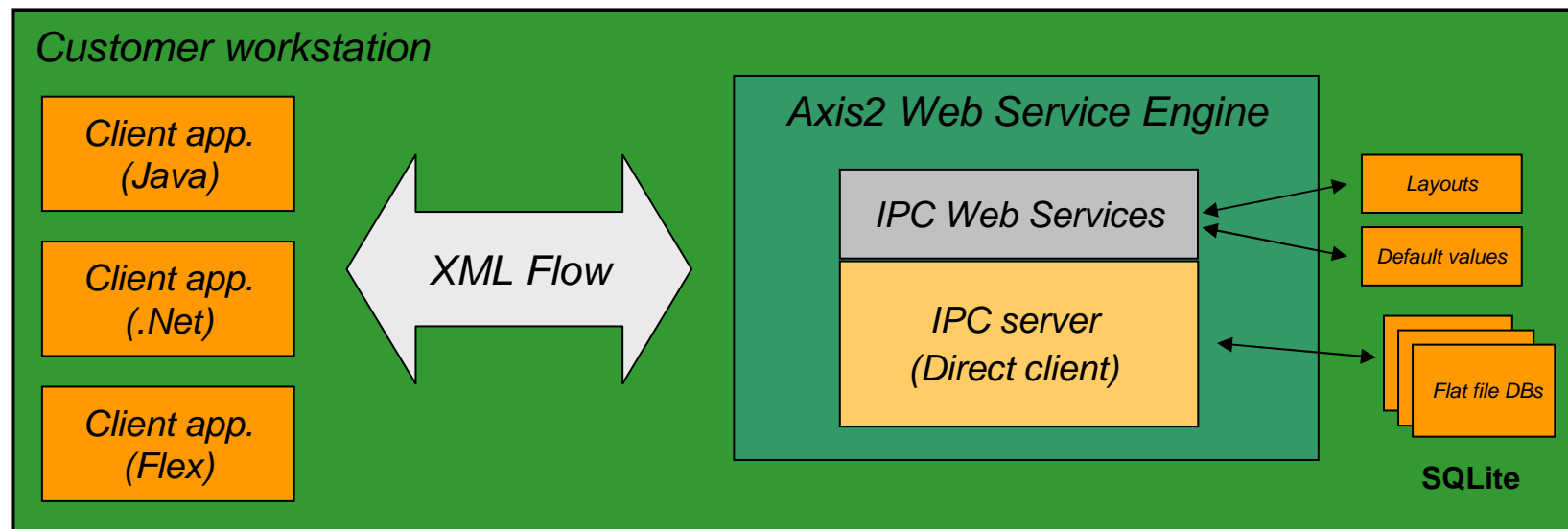
# Online IPC for remote users

- IPC Standalone + Data Loader
- Browser based client applications in Flex, Ajax, etc ...
- Standalone client applications in .Net, Java, Air, etc ... with remote access to IPC



# Standalone IPC on customer workstation

- IPC Standalone + flat file KB import
- Installation should be as easy as possible
- Easy IPC database distribution and update with limited cost → **File based DB : SQLite**
- Client applications of IPC are written in JAVA, .Net, Flex ... so we need a language independent interface for IPC → **Web Services**



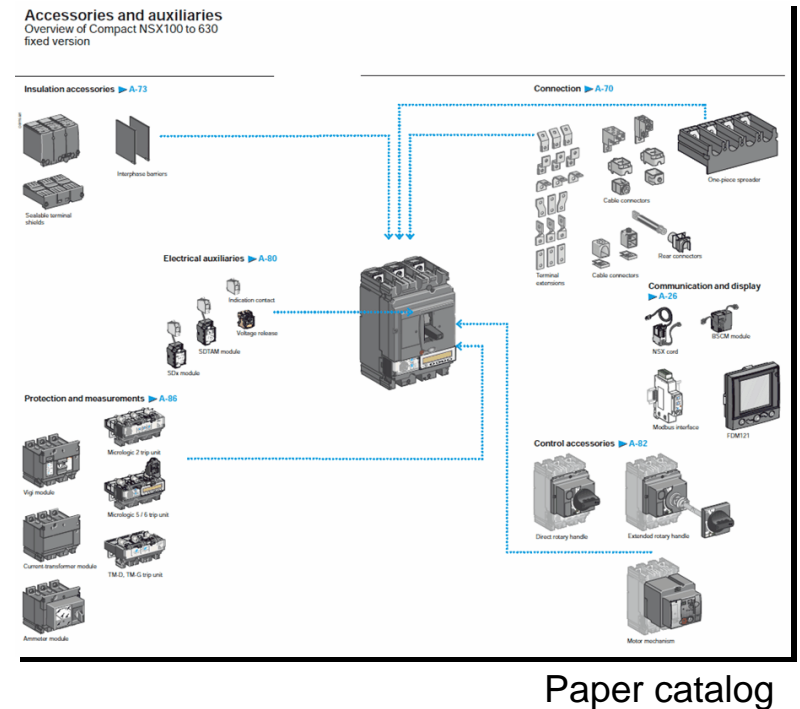
# Common, language independent interface to IPC

- **XML based interface via Web Services, no matter which client as long as it supports Web Services**
- **The Web Service layer is the same for online and offline use**
  - Reduced maintenance cost
- **IPC behavior can be enhanced in the Web Service layer:**
  - Default values management
  - Presentation layouts ( reordered cstic value list, cstic groups and subgroups, simple or expert mode configuration, ...)



# Sales oriented graphical UI

- Is used to promote new products on the internet
- Graphical representation close to the paper catalog version
- Simple and user friendly graphical UI for complex products
- Runs on a web browser with Adobe Flash Player



*Demo ...*

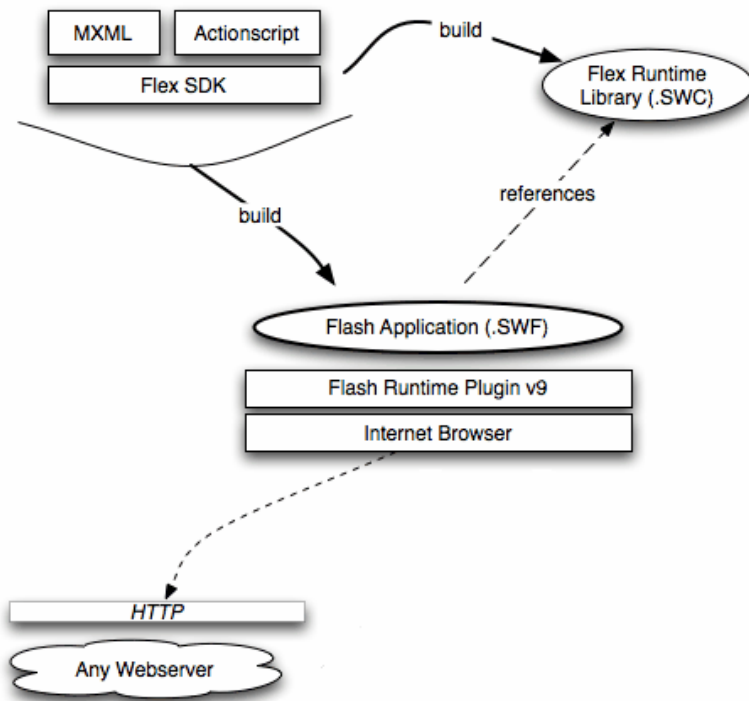
# The configuration model

- Has over 800 cistics
- Multi-level (up to 7) with 350+ configurable items in the static BOM
- Only one material used for sales and production

Plant=0001 BOMAppI=SD01 Date=200904...	Descr.
▼ GCR_NS100	Material NS100 (1310)
▼ GCR_COMNN13079	Material NS100-630 (
▶ GCR_COMCD13042	Complete Device (130
• GCRS_D13081_MG4	GCRS_D13081_MG4
▼ GCR_COMTU13082	Trip unit (13082)
• GCRS_D13082_MG4	GCRS_D13082_MG4
• GCRS_D13513_MG4	GCRS_D13513_MG4
• GCRS_D13514_MG4	GCRS_D13514_MG4
▼ GCR_COMEO13512	Electronic trip unit
▶ GCR_COMNO13502	NS trip unit options
▶ GCR_COMCO13511	C trip unit options
• GCRS_D93042_MG4	GCRS_D93042_MG4
• GCRS_D13083_MG4	GCRS_D13083_MG4
▼ GCR_COMMM13111	Motor mechanisms SD
• GCRS_D13111_MG4	GCRS_D13111_MG4
• GCRS_D13043_MG4	GCRS_D13043_MG4
• GCRS_D13044_MG4	GCRS_D13044_MG4
• GCRS_D26907_MGD	GCRS_D26907_MGD
• GCRS_D26910_MGD	GCRS_D26910_MGD
▶ GCR_COMRH13098	Rotary handles (1309
▶ GCR_COMEA13052	Electrical auxiliari



# Adobe Flex



- Flex™ is a highly productive, free open source framework for building and maintaining expressive web applications
- Flex applications can be deployed on all major browsers, desktops, and operating systems
- MXML is a declarative XML-based language used to describe UI layout and behaviors
- ActionScript™ 3 is an object-oriented programming language used to create client logic
- Flex includes a rich component library with more than 100 proven, extensible UI components for creating Rich Internet Applications (RIAs), as well as an interactive Flex application debugger

# Generic UI with default values management

**BOM**


- ▼ GCR\_NSX100\_630
  - ▼ GCRI\_NSXTU13275
    - GCRS\_C13274\_NSX
    - GCRS\_C13273\_NSX
  - ▼ GCRI\_NSXV13412\_
    - GCRS\_C13412\_NSX


**CSTICS** GCR\_NSX100\_630/

- breaking capacity: 100\_KA\_380\_415\_AC\_00
- circuit breaker name: Compact NSX400S
- In rated current: 400\_A
- circuit breaker rating code: S
- control type: toggle
- Vigi: with
- poles description: 3P
- mounting mode: fixed
- Display communication: no
- Safety signalling control: no
- indication and measurement mod: no
- Connection accessories: no
- locking device: no
- installation accessories: no
- individual enclosures: no

**SUMMARY**

Instance id	Cstic	Value
▼ GCRI_NSXTU13275	Product reference	LV432414
	circuit breaker nar	Compact NSX400S
	circuit breaker rati	S
	poles description	3P
	In rated current	400_A
	breaking capacity	100_KA_380_415_AC_00
▼ GCRI_NSXV13412_	range of product	NSX400...630
	poles description	3P
	protected poles d	3t
▼ GCR_NSX100_630	breaking capacity	100_KA_380_415_AC_00
	circuit breaker nar	Compact NSX400S
	In rated current	400_A
	circuit breaker rati	S
	control type	toggle
	Vigi	with
	poles description	3P
	mounting mode	fixed
	Display communic	no
	Safety signalling c	no
	indication and me	no
	Connection acces	no
	locking device	no
	installation access	no
	individual enclosu	no





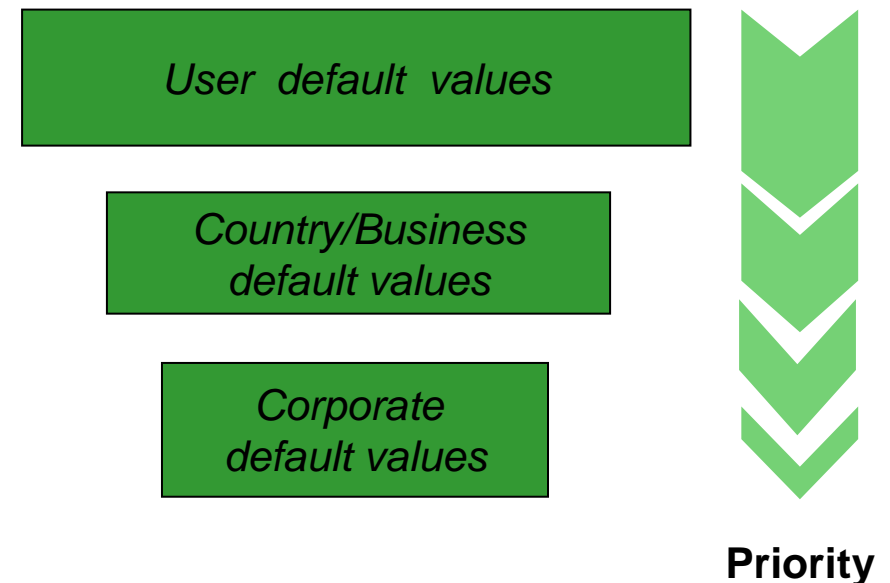
# Requirements

- **At the beginning of the configuration session the user should see values for every cstic**
  - each cstic should have a default value
- **If he's ok the user can leave the configuration session and the default values will be applied**
- **If not, the user can change some cstic values knowing that default values do not restrict any cstic value domain**
- **During the configuration, the compatible default values with the user set cstics should continue to show up**
- **Default values for a multi-level model are stored in an ordered list (in un XML file)**
- **Default values must be maintained easily by different actors: central, country, business/software, user.**

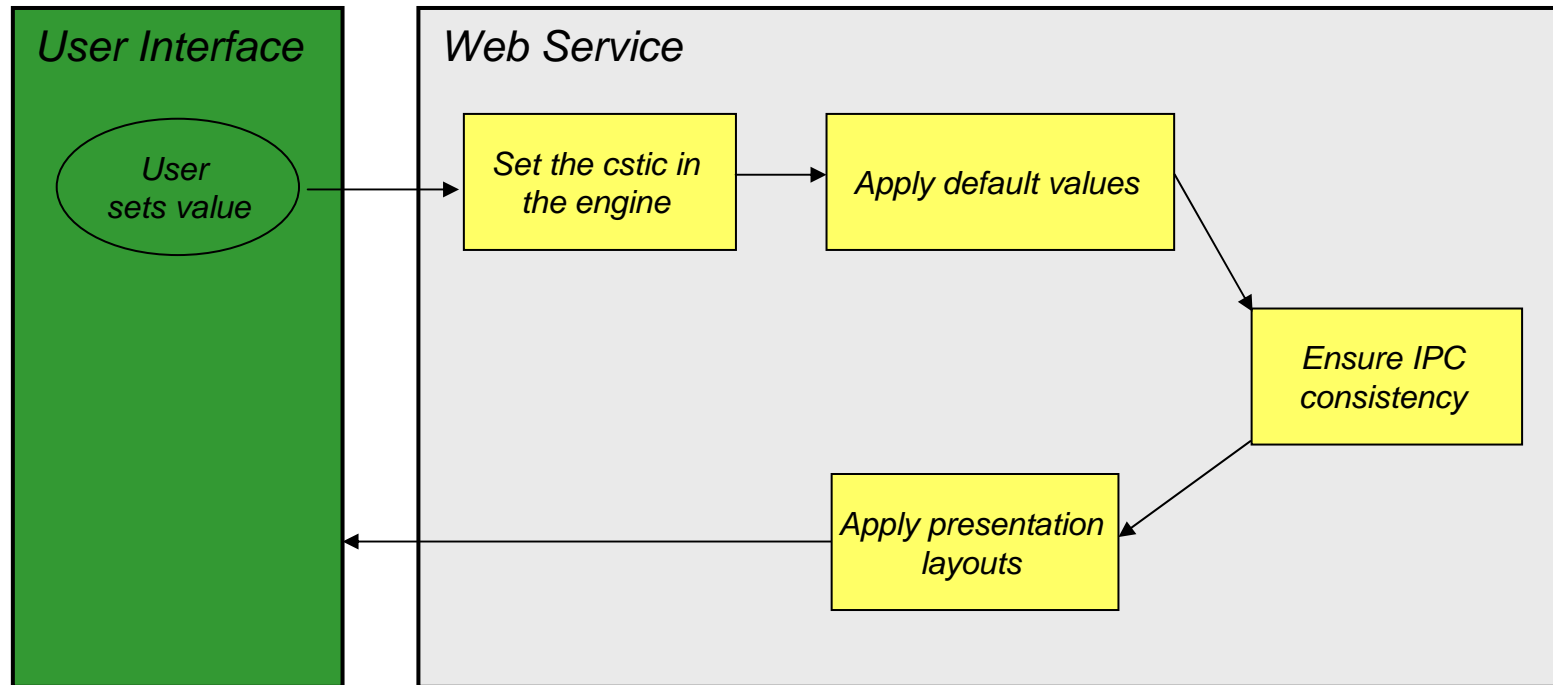
***Demo ...***

# Default values definition

- Default values are defined by each Schneider actor following the context and the usage
- The user can also define his own default values
- The solution should reduce model maintenance tasks



# Default values calculation



# Why use custom default values management ?

## ■ SAP default values limitation

- Default values restrict other cstic value domains
- Default values cannot by easily prioritized
- Default values are managed by modeling: cannot have user level default values
- Default values are managed by modeling: requires to rebuild KBs when default values are created or updated

# Presentation layouts

- **Cstic and cstic value display reordering**
- **Extended cstic display options (list box, check box, hierarchical values, image ....)**
- **Cstic and value description substitutions (30 char+)**
- **Unlimited number of languages/dialects supported**
- **Cstic grouping/sub-grouping**

# Summary

- **Use of Flex for enhanced configuration UIs**
- **Web Services for language independent and unified access to IPC, either online or offline**
- **Use of flat file database for easy distribution of IPC, embedded in customer software**
- **Custom default values management system**
- **Presentation layouts for easy customizing and localizing model behavior**