

SCENARI3

Stéphane Crozat



epicure
Journée Bilan



SCENARI3

- ❖ SCENARI3 désigne l'ensemble du code informatique sous licence libre, gratuite et open source résultat du projet Epicure
- ❖ Caractéristiques techniques principales :
 - Intégration multimédia (audio)
 - Champs d'usage divers (webradio, présentation de PME, etc.)
 - **Création d'applications sans développement informatique (SCENARIbuilder)**

Avant Epicure : SCENARchain2

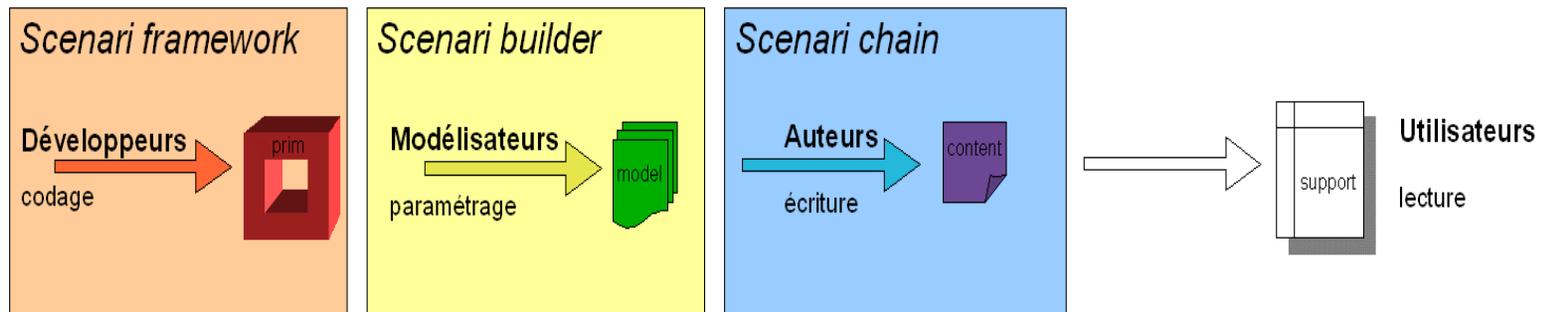
- ❖ Environnement de création et de publication de contenu
- ❖ Programmé par un **modèle** documentaire et pédagogique
 - Schéma documentaire
 - Éditeur graphique
 - Règles de gestion
 - Moteurs de publication

SCENARbuilder

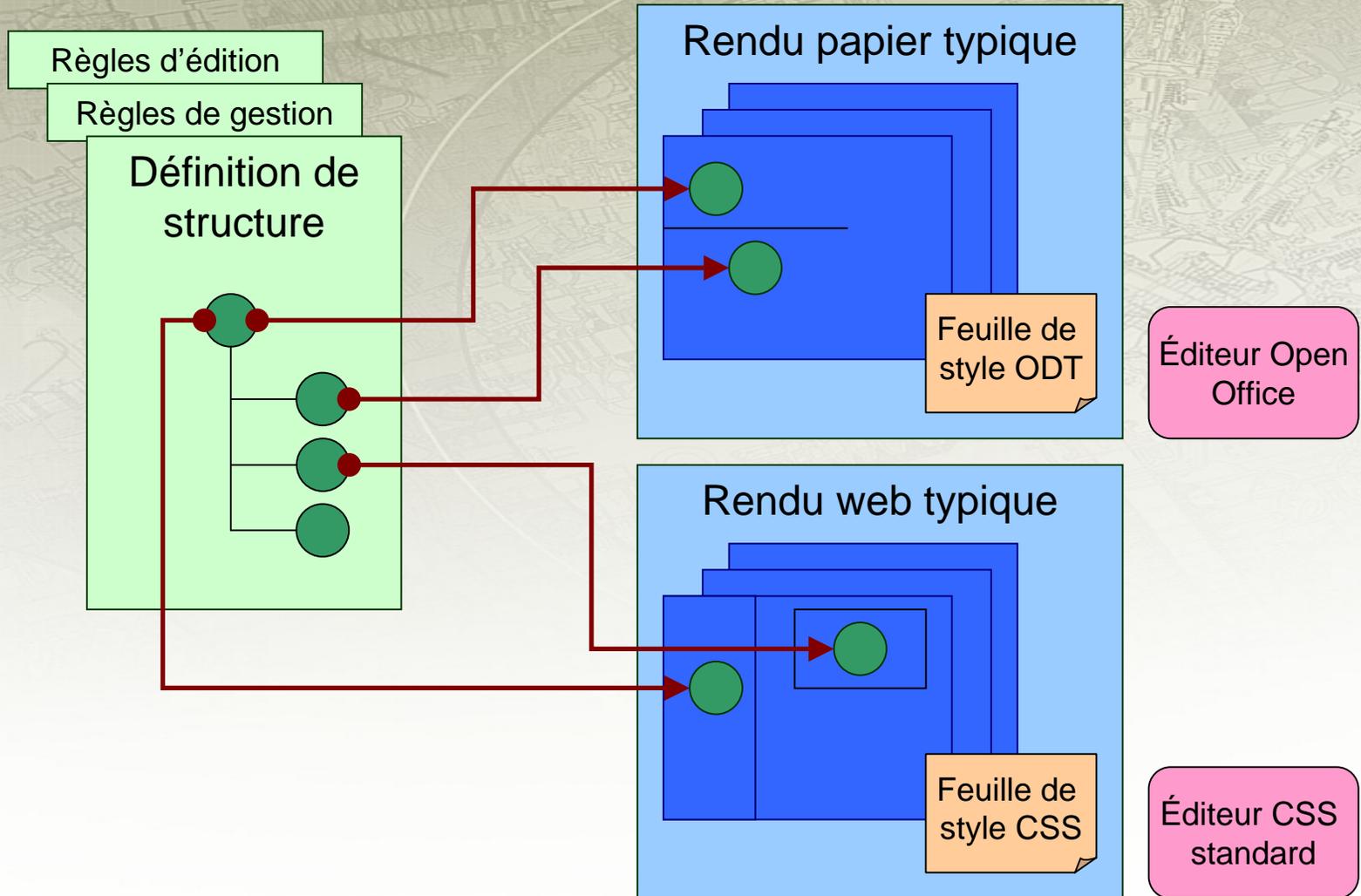
- ❖ Un outil de création de chaînes éditoriales par paramétrage
- ❖ Les modèles issus de SCENARbuilder sont installés dans SCENARchain3
 - Ils peuvent à présent être créés intégralement par les modélisateurs (non informaticiens)

Processus SCENARI3

- ❖ A partir de SCENARIframework (ensemble de primitives existant à un instant t)...
- ❖ ... le modélisateur crée un ensemble de modèles (paramétrage de primitives) dans SCENARibuilder et exporte une définition d'atelier ...
- ❖ ... l'auteur crée le contenu dans SCENARichain3 dans le cadre de l'atelier ainsi défini



Principe du paramétrage



Démonstration

Epicure
Journée Bilan



```
sm:compositionPrim name="Présentation simple" category="" info=""
```

```
sm:help quickHelp="Permet de scénariser le contenu de la présentation avec un seul niveau de plan"
```

```
sm:structure
```

```
sm:meta sc:refUri="informations.model" usage="required"
```

```
* sm:part code="logo" name="Logo" family="content" internalized="never" usage="optional"
```

```
* sm:allowedModel sc:refUri="gif.model (/utc/sc3/resources)"
```

```
* sm:allowedModel sc:refUri="jpeg.model (/utc/sc3/resources)"
```

```
* sm:allowedModel sc:refUri="png.model (/utc/sc3/resources)"
```

```
* sm:part code="preamble" name="Préambule" family="" internalized="always" usage="required"
```

```
* sm:meta sc:refUri="title.model (/utc/sc3/data)" usage="required"
```

```
* sm:allowedModel sc:refUri="contenu.model (/showOrg/itemsCommuns)"
```

```
* sm:set usage="required"
```

```
* sm:part code="section" name="Section" family="" internalized="always"
```

```
* sm:meta sc:refUri="title.model (/utc/sc3/data)" usage="required"
```

```
* sm:allowedModel sc:refUri="contenu.model (/showOrg/itemsCommuns)"
```

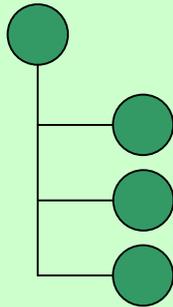
```
*
```

Principe du paramétrage

Règles d'édition

Règles de gestion

Définition de
structure





```
sm:compositionPrim name="Exemple" category="Intention" info=""
```

```
sm:help...
```

sm:structure

```
sm:meta sc:refUri="🔧 title.model (/utc/sc3/data)" usage="📄 optional"
```

```
* sm:part code="content" name="Contenu" family="📄 content" internalized="
```

```
* sm:allowedModel sc:refUri="🔧 resources.model (/academic/Re)"
```

```
*
```

```
*
```

sm:authoring

```
sm:icon sc:refUri="📁 "
```

```
sm:itemNameCreator...
```

```
sm:formEditor...
```

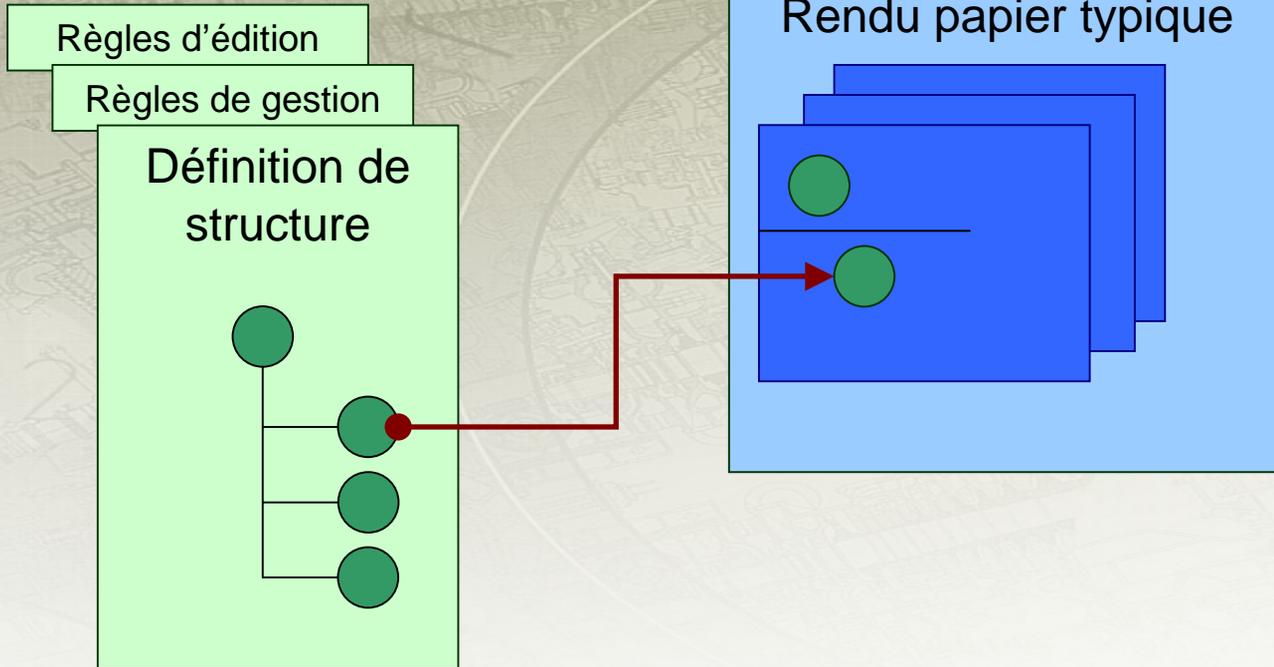
sm:htmlPreview

```
* sm:block refCode="content"
```

```
sm:iconBlock sc:refUri="🖼️ example.png (/academic/img)"
```

```
*
```

Principe du paramétrage





sm:compositionOdtTransf info=""

sm:model sc:refUri="example.model (/academic/Pb)"

sm:axis code=""

sm:content

* sm:for codes="content"

* sm:WPictureBlock style=""

sm:title

* sm:partTitle prefix="" suffix="" format="fullTitleText"

* sm:fixedTitle value=""

sm:picture sc:refUri="exemple.jpg (/academic/generators/od_

* sm:callSubModel axis=""

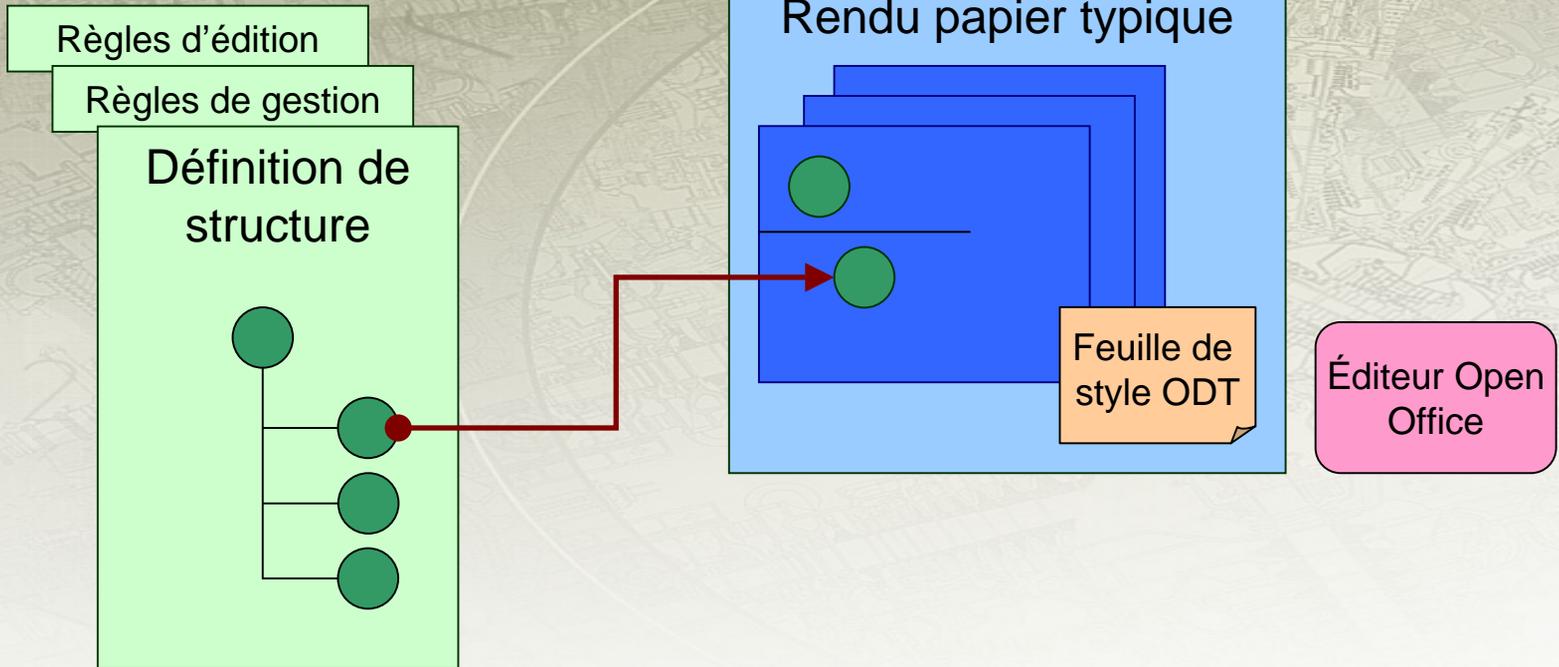
* sm:defaultStyle...

*

*



Principe du paramétrage



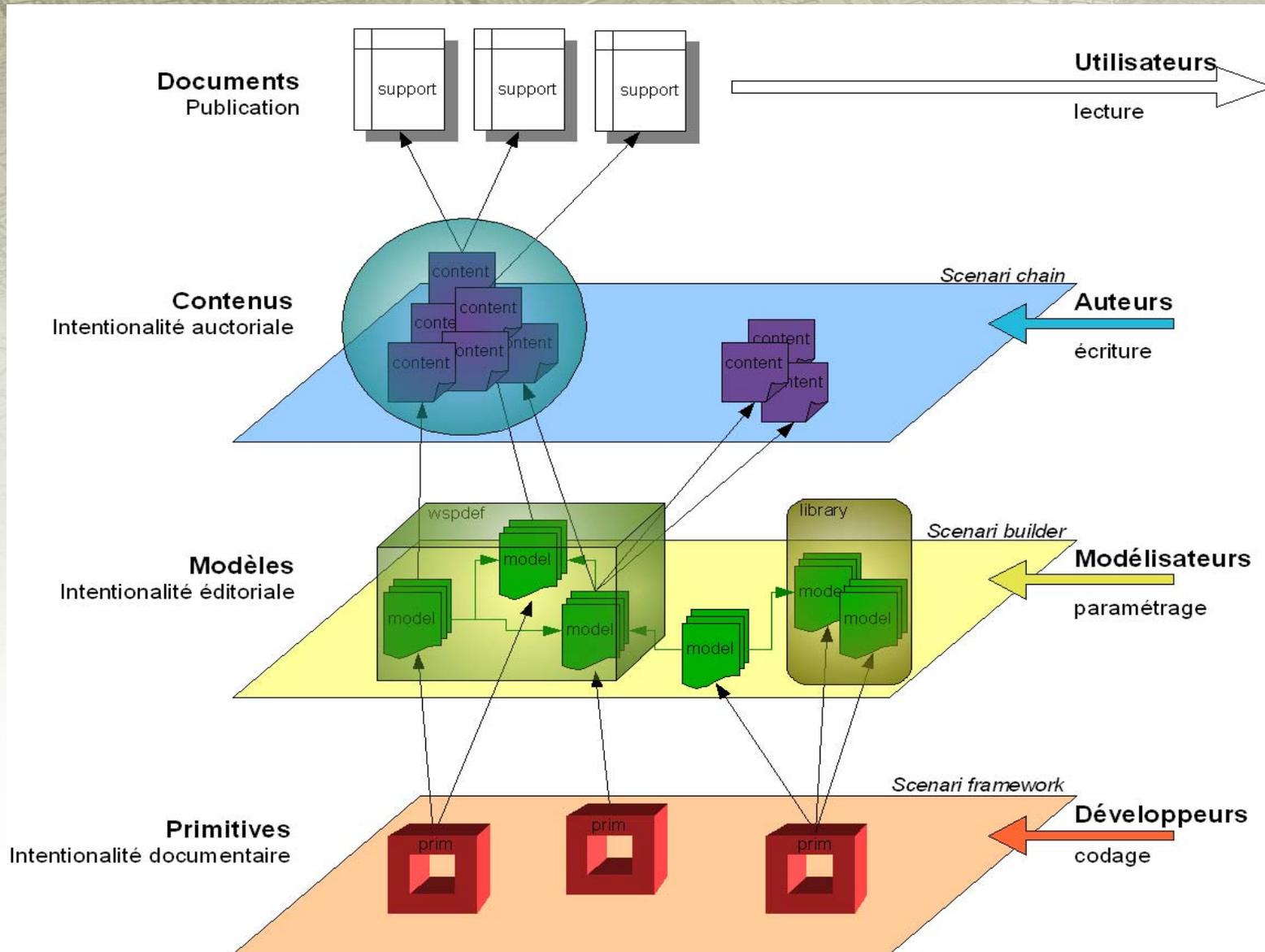
Technologies exploitées

- ❖ Les interfaces graphique d'édition et de gestion
 - Technologie Mozilla
 - Extension « XED »
 - Technologies XML-XSL-RelaxNG
- ❖ Le cœur applicatif de gestion et publication
 - Technologie Java
 - Technologies XML-XSL
 - Extension « Feuilles de Comportement »
- ❖ Les formats de publication
 - Technologies Web (HTML, JavaScript, Flash)
 - Technologies Open Office

Prestataires « exploités »

- ❖ Kelis conseil & développement
 - Architecture, développement du framework, moteurs de publication Open Document et HTML
- ❖ Disruptive Innovations
 - Expertise Mozilla et intégration audio
- ❖ Loic Alejandro
 - Publication HTML
- ❖ Gabriel Leroy
 - Moteur de publication Flash

Architecture SCENARI3



Pour conclure : une « autre » généricité

- ❖ Passage de la logique classique modèle-instance...
- ❖ ... à une architecture plus riche :
 - Programmer le spécifique en créant des primitives dans le framework
 - Logique de programmation
 - Paramétrer le spécifique à partir des primitives disponibles
 - Logique de déclaration
 - Décliner le spécifique à partir de bibliothèques de réponses typiques
 - Logique de réutilisation/adaptation