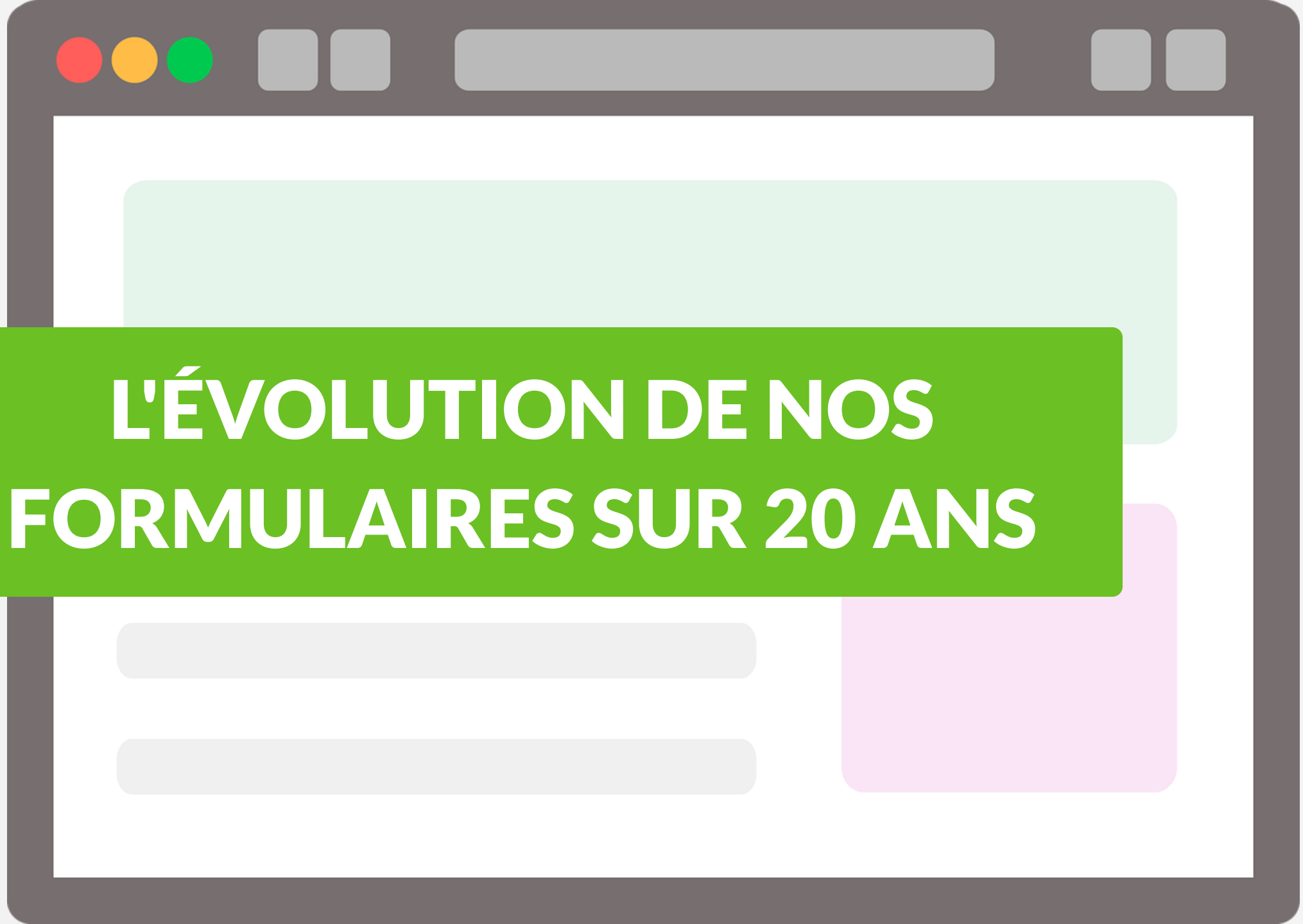
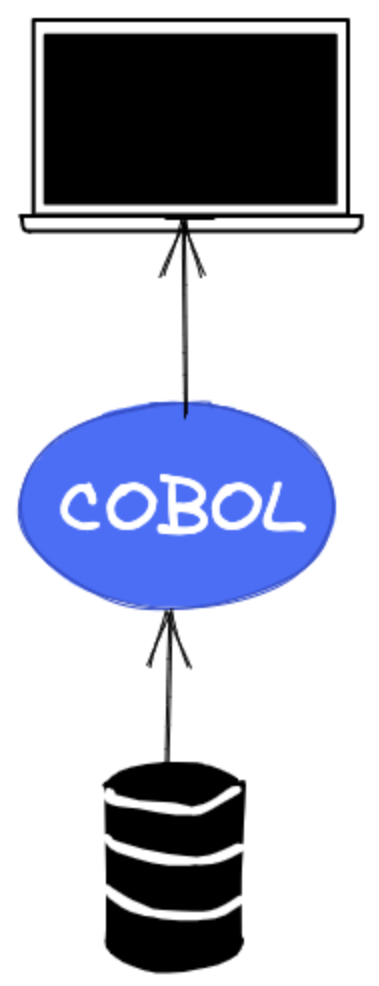


L'ÉVOLUTION DE NOS FORMULAIRES SUR 20 ANS

QUI JE SUIS ?

Christophe Genin

- Référent technique et fonctionnel à la mutuelle de poitiers assurance.
- Vieux dev fullstack ^^
- twitter : @skarboune



Et au début était

COBOL

Par contre comment faire interragir les utilisateurs ?

Création d'une application desktop via powerbuilder

Année 2000

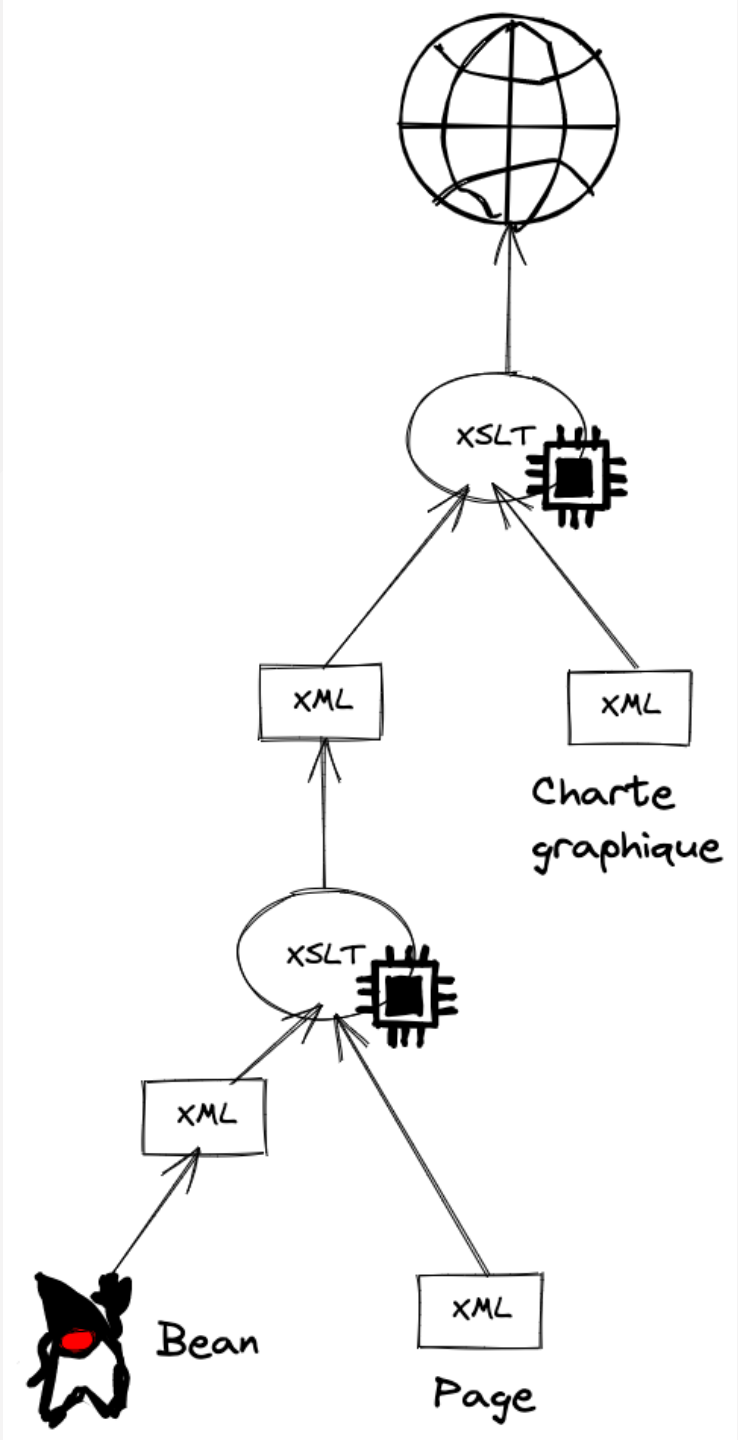
La révolution du web

Monolithe Java

Basé sur Struts.

Mais IHM basé sur **XSLT**

- Page web définit par un fichier XML.
- Une charte graphique qui transforme le XML en HTML.



```
<tableau type="resultat">
    <entete>
        <colonne tri="texte" intitule="" largeur="5%" classe="aligneCentreH" />
        <colonne tri="texte" intitule="Date de création" largeur="10%"
            classe="aligneCentreH" />

        <colonne tri="texte" intitule="Internaute" largeur="15%"
            classe="aligneCentreH" />
        <colonne tri="texte" intitule="Agent" largeur="17%"
            classe="aligneCentreH" />
        <colonne tri="texte" intitule="Demande" largeur="23%"
            classe="aligneCentreH" />
        <colonne tri="texte" intitule="Etat" largeur="20%"
            classe="aligneCentreH" />
        <colonne tri="texte" intitule="Actions" largeur="10%"
            classe="aligneCentreH" />

    </entete>
    <xsl:for-each select="/mdpa/formulaire/listeActions/item">
        <ligne>

            <cellule largeur="5%">
                <xsl:choose>
                    <xsl:when test="enRetard='true'">
                        <image aligner="left" infobulle="La demande est en retard de traitement."
                            url="/_images/warningError.gif" hauteur="16" largeur="16" />
                    </xsl:when>
                    <xsl:otherwise>
                        <vide></vide>
                    </xsl:otherwise>
                </xsl:choose>
            </cellule>
            <cellule largeur="19%">
                <constante
                    valeur="{dtCreation/dateFormatee} {dtCreation/heureFormatee}" />
            </cellule>
```

	Date de création	Internaute	Agent	Demande
	27/12/2021 17:52	[illegible]	8050-0	Conversation
	27/12/2021 17:32	non identifié	3800-0	LeLynx devis
	27/12/2021 17:15	[illegible]	1420-0	Conversation
	27/12/2021 17:09	[illegible]	2670-0	Conversation
	27/12/2021 17:05	[illegible]	8050-0	Conversation
	27/12/2021 16:43	[illegible]	0130-0	Attestation

Bilan de cette première étape

- 😄 Librairie graphique basée sur des composants ➡️ 🚀
- 😕 Pas d'interaction possible avec le Html.
- 😕 Langage non intuitif et difficile d'accès.

Mais surtout :

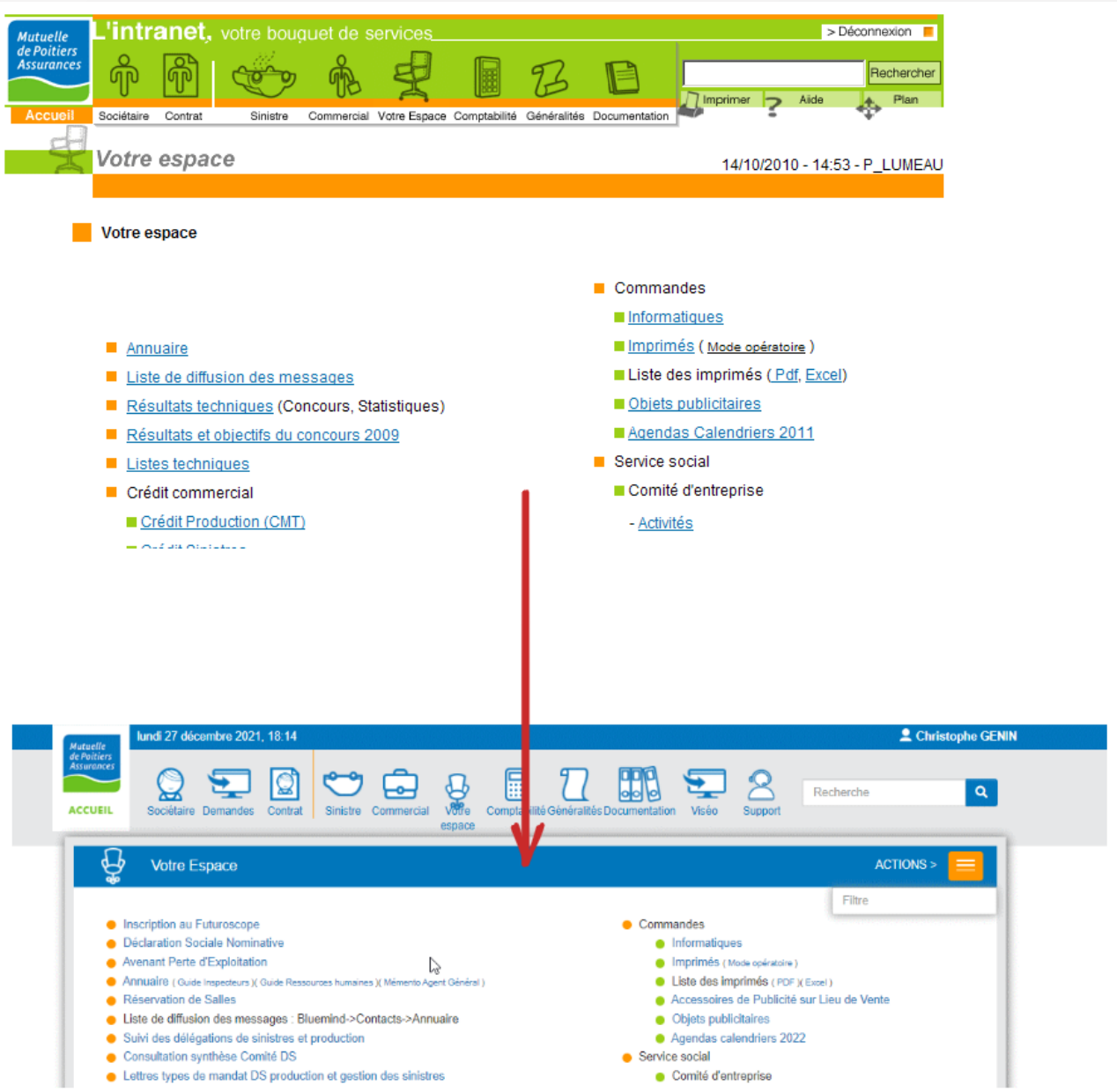
Librairie mal construite : un composant peut générer plusieurs éléments différents

```
Un composant doit avoir une granularité fine et être pur.
```

Framework basé sur flash et possédant une intégration avec Spring.

- ☹ Framework avec des bugs
- ☹ Temps de développement assez long des ihms.
- ☹ Projet POC important en temps de développement.

```
Tester vos nouvelles briques sur des micro projets
non essentiel à votre business.
```



Le changement de charte graphique

Effet de bord : Les utilisateurs pensent que ce sont de nouvelles applications **et Trouvent des bugs imaginaires**.

```
En cas de gros changements graphiques,
prévenez vos équipes support !!
```

Architecture Micro services

Décision de mettre en place une architecture micro services et de créer vers des ihms créés en JS.

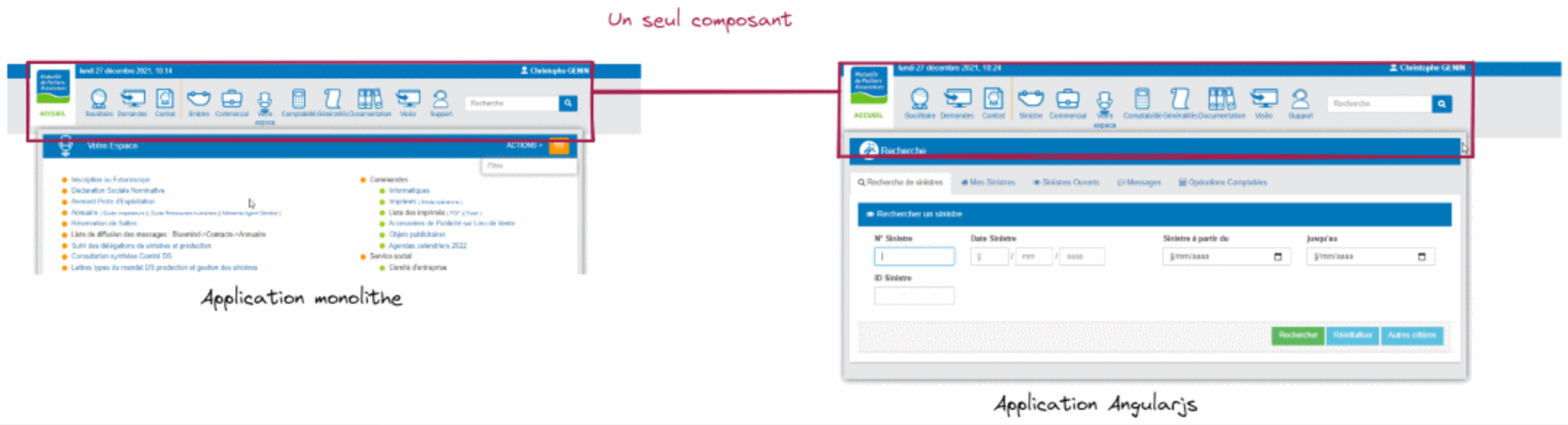
Pour rappel, l'ancètre de Angular.

Mais, comment valider cette brique ?

Le composant menu

Pré-requis : Avoir nos futures applications ayant le même rendu que celle du monolithe.

💡 **Remplacer le menu du monolithe par un composant javascript.**

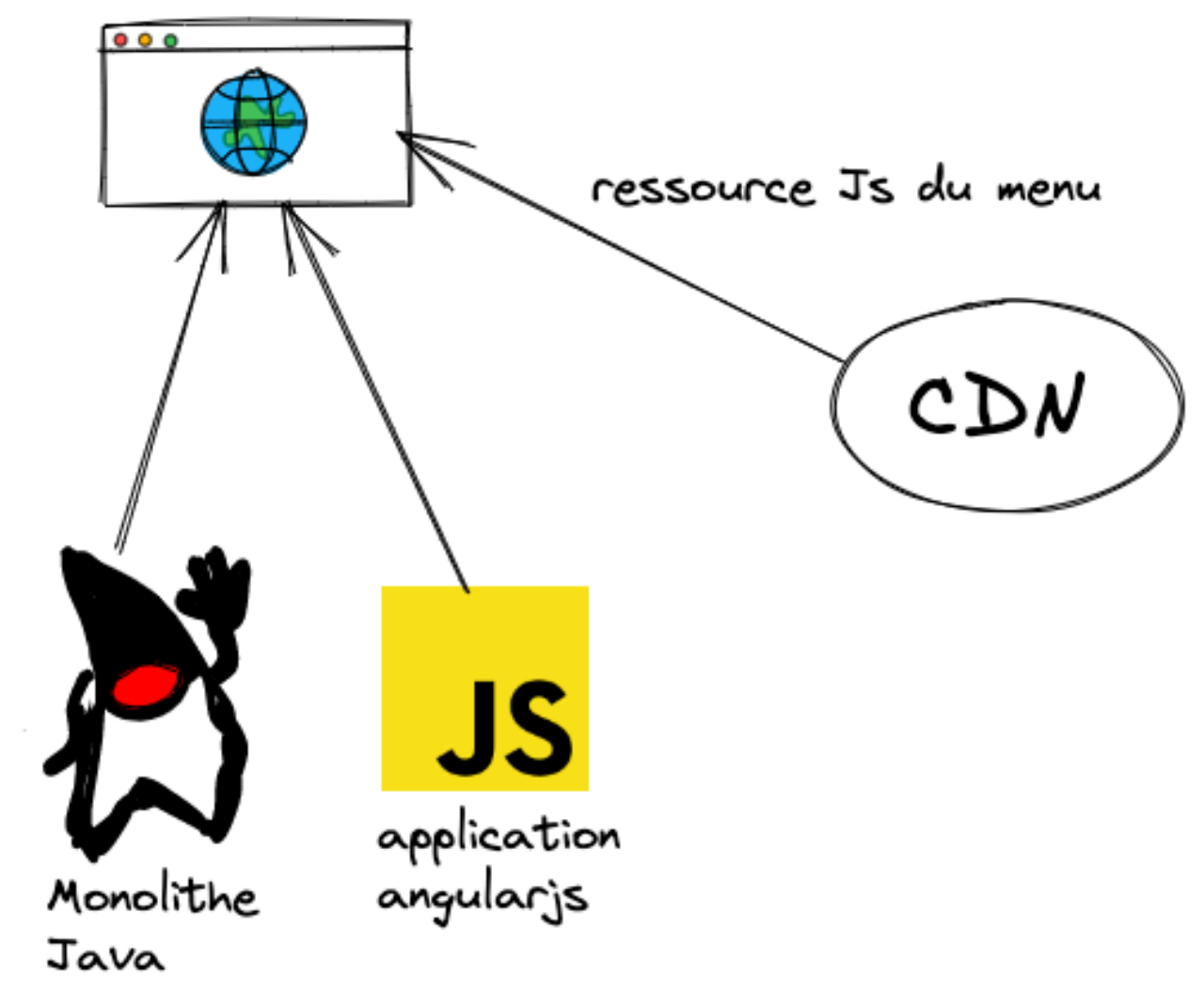


Fail : les mise à jour du menu.

Première version : inclusion du code Js dans chacun des projets.

➡️ Maintenance difficiles.

💡 **Définir un cdn interne.**



Pas de centre de gestion.

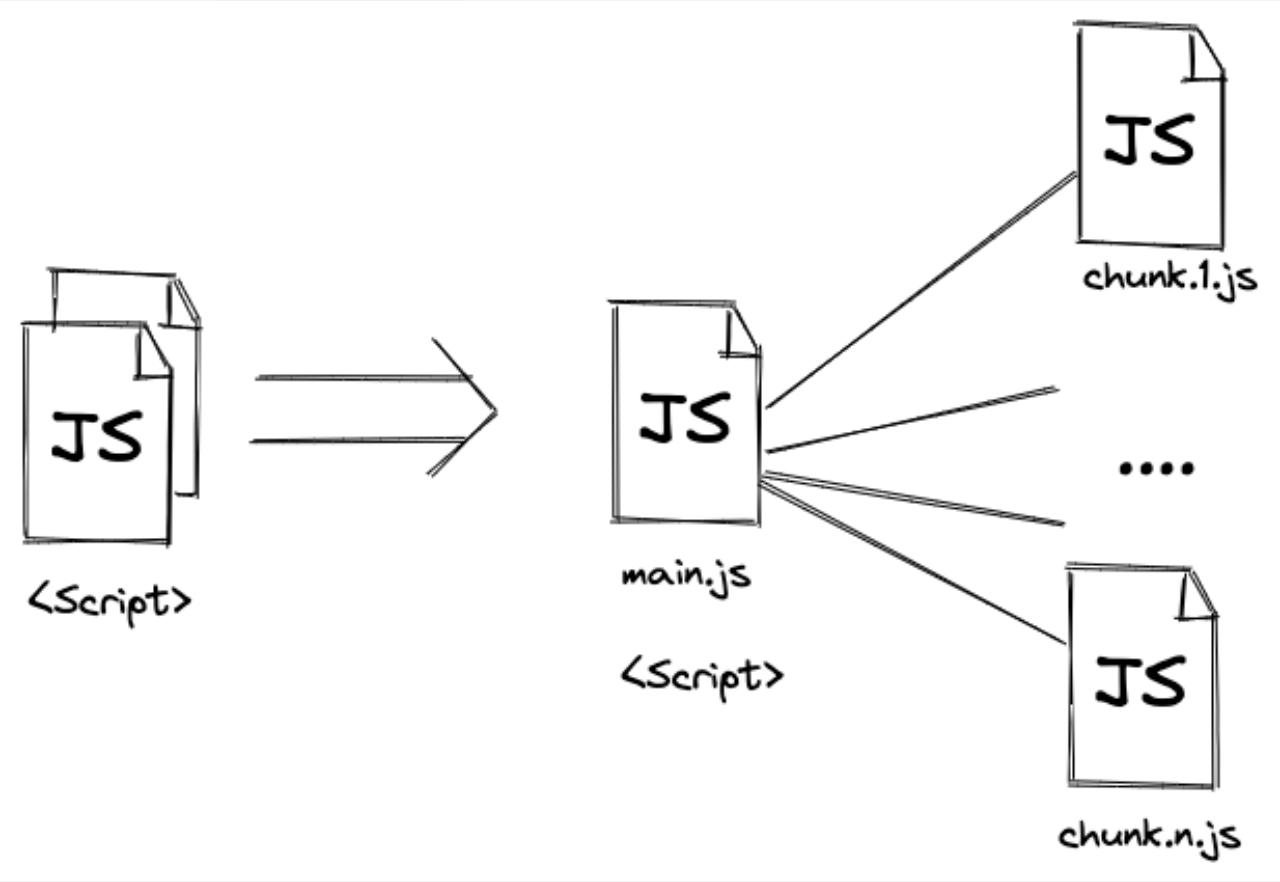
Utilisateur n'ayant pas forcément un réseau fibre...

```
Solution : Cacher vos fichiers de ressources via http.
```

D'autres amélioration

- Le code splitting
- Et d'autres évolutions ....

```
Toujours se garder
informer des nouveautés
```



Vers un nouveau framework ?

Décision de la DSI : Utiliser les mêmes technologies dans la partie internet.

Et là, Angularjs ne fait plus le job.

Les développeurs sont satisfaits d'angularjs pour les applications internes. => 🏆 VueJs est le gagnant.

```
import template from "./../template/AjouterBlocNotes.html";


class AjouterBlocNoteCtrl {
    constructor($state, $filter, noteService, acteurService, sinistreService) {
        this.$state = $state;
        this.$filter = $filter;
        this.noteService = noteService;
        this.acteurService = acteurService;
        this.sinistreService = sinistreService;
        const now = new Date();
        this.date = new Date(now.getFullYear(), now.getMonth(), now.getDate(), now.getHours(), now.getMinutes());
        this.subject = '';
        this.visibility = 'T';
        this.body = '';
    }

    validerNote(note) {
        this.noteService.add(note);
        this.$state.go('sinistre.process.gestion.blocnotes.lister', {
            id: this.sinistreService.ctx.sinistre.id
        },{reload : true});
    };

    annulerNote() {
        this.$state.go('sinistre.process.gestion.blocnotes.lister', {
            id: this.sinistreService.ctx.sinistre.id
        });
    }
}

AjouterBlocNoteCtrl.$inject = ['$state', '$filter', 'noteService', 'acteurService', 'sinistreService'];

export default {
    name: 'ajouterBlocNote',
```

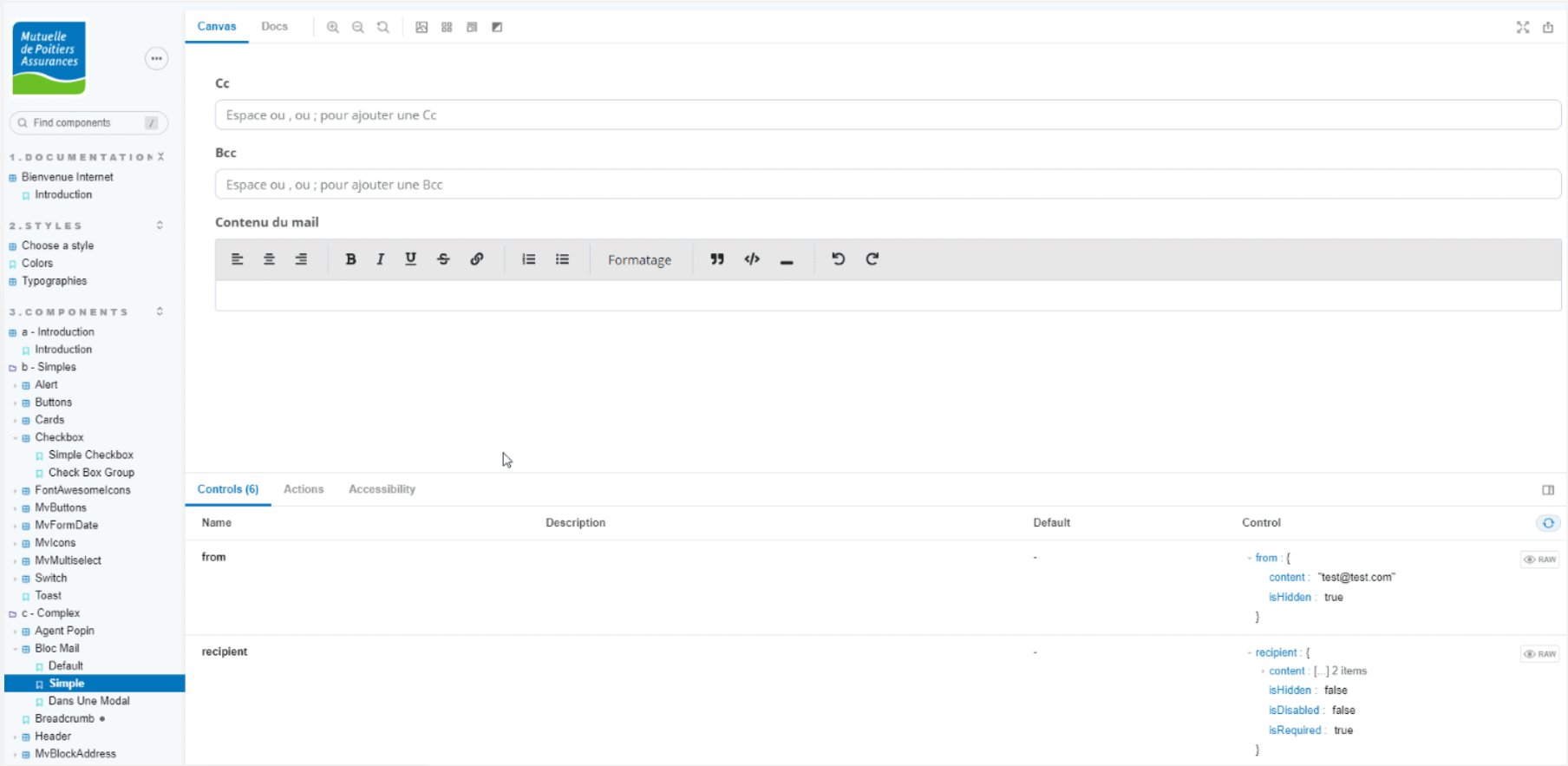
Angularjs

```
<template>
  <b-row class="mt-4 mb-3">
    <b-col class="text-center">
      <h1 class="text-primary font-size-xxl font-weight-normal" v-html="titleCo
      <p v-if="info" class="text-gray-dark font-size-xs font-italic">{{ info }}
    </b-col>
  </b-row>
</template>
<script lang="ts">
import ...

@Component( options: {
  components: {BCol, BRow}
})
/**
 * AppTitle description and version
 */
export default class AppTitle extends Vue {
  @Prop( options: {type: String, required: true}) titleContent!: string;
  @Prop( options: {type: String}) info!: string;

}
</script>
```

Vuejs

Bibliothèque de composants

Utiliser Storybook



Ne pas oublier le passé

Architecture Decision Record : ADR

Effectuez des comptes rendu de vos décisions d'architecture même celles impactant vos IHMs.

Utilisez vos wikis, vos issues, vos outils de documentations projets pour noter vos essais, vos échecs, etc...

Et surtout faîtes qu'elles soient accessibles à tous vos développeurs.

En conclusion

- L'utilisation de composants permet de créer des applications de gestion complexe rapidement.
- Testez vos nouvelles briques sur des périmètres restreints (N'oubliez pas votre support et vos ops).
- Attention les nombres d'incidents augmentent en cas de gros changements graphiques.
- Ayez une bonne connaissance des bases (Protocole Http).
- Restez en veille (Sinon vous êtes venu pour la soirée vin rouge fromage ?).

- Collaborez et écoutez les remarques de toutes les personnes de vos DSI (Prestas inclus).
- Notez vos décisions, vos succés et vos échecs.

Merci de m'avoir écouté