

Web Designer

Constat

- Web peu efficace
 - Performance
 - Esthétiquement limité (complexité) (*de base*)
 - Développement lent et instable
- Nécessité de « simplifier » le web
 - Développement plus simple/plus rapide
 - Ne plus réinventer la roue
 - Créer des outils/environnements tout-en-un

Réponse peu satisfaisante

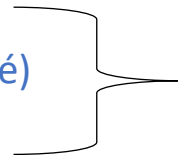
- Réponse actuelle

- Développement des Frameworks Front-End (simplifier le Front-End)
- Popularisation de NodeJS (« unification » par le Front-End des langages)
 - ≈ faire disparaître/cacher le back-end et ses contraintes (performances, testing, contraintes réseaux, etc.)



- Web peu efficace

- Performance
- Esthétiquement limité (complexité)
- Développement lent et instable



Résolution **théorique** de seulement **la moitié des problèmes...**

Réponse peu satisfaisante

- **Problème de performance :**

« *Comment en une seule seconde, je peux produire 60 images de 8 millions de pixels, et ne pas réussir à charger une page web de pur texte dans un navigateur ?* »

- Problématiques réseaux abstraites du développeur
- Pas d'algorithmique – Pas de théorisation
 - Décorrélation entre la réalité physique de l'ordinateur et le développement du logiciel (*Code pas cache-friendly par exemple*)
 - Peu ou pas d'optimisation (*surutilisation du cache du navigateur, taille mémoire, exécution du code, etc.*)
- Mentalité Consumériste/Je-m'en-foutiste : « J'ai de la ressource, autant l'utiliser ! »
 - Présuppose que tout le monde possède une machine surpuissante et ne va que sur un seul site
 - Posture sociale et idéologique :
 - « Aujourd'hui on est passé à autre chose : s'occuper des performances = ne pas s'occuper du design » (*aka ne pas être moderne*)
 - « Les performances c'est bon pour les devs C des années 1980 qui utilisent des patates comme PC : aujourd'hui, tout le monde a une bonne machine »

Réponse peu satisfaisante

- **Problème d'esthétique:**

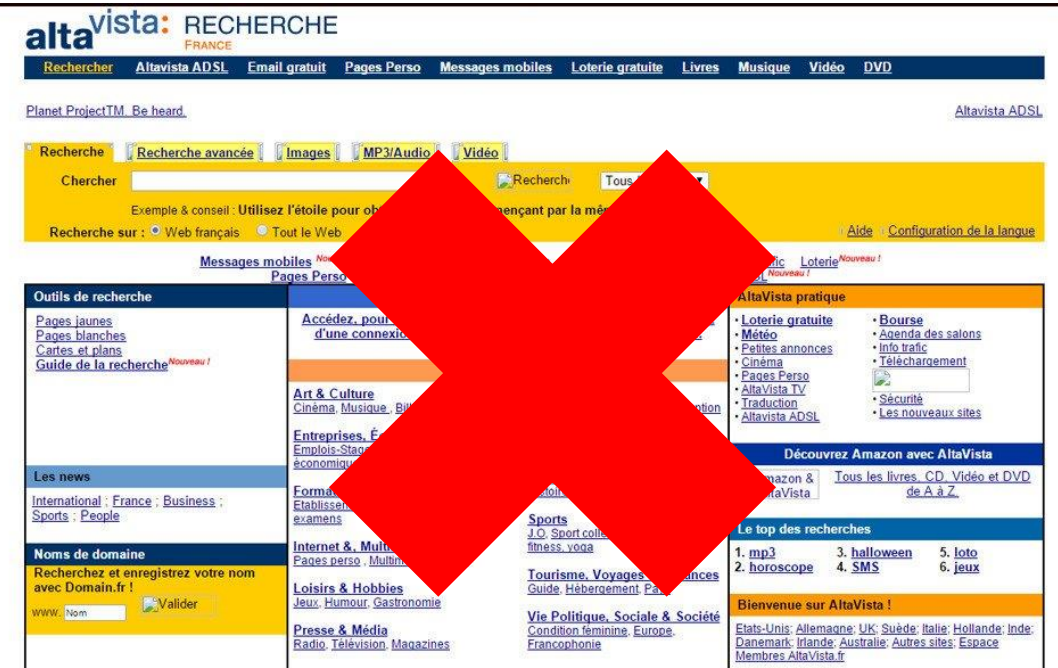
« *Comment faire en sorte que mon site ne ressemble pas à une bouse des années 1990-2000 ?* »

- **C'est un faux problème**

- Ça n'est qu'une question de design, de couleurs, etc.
- Aujourd'hui il n'y a pas de limitation techniques

- **Réel problème**

- Complexité du CSS (Positionnement, Gestion des styles)
 - Instabilité du CSS (Erreurs silencieuses)
 - Peu d'évolutions marquantes du CSS
- CSS = un langage de développeurs pas de designer



Réponse peu satisfaisante

- **Problème de vitesse de développement :**

« *Pourquoi faut-il si longtemps pour développer un site web et si peu pour en avoir l'idée ?* »

- Redondance forte / Peu d'automatisation

- Écriture du **HTML** *difficilement automatisable* (faisable mais souvent plus long que de l'écrire)
 - **CSS** *langage peu formel* (encourage la redondance « au cas où »)
 - Il faut **coder chaque comportement manuellement**
- C'est comme demander au menuisier d'expliquer à chaque client comment utiliser une porte

- **Mélange des compétences**

- Mix entre compétence techniques (développement, logique) et artistiques (design, mise en forme, UX)
- Compétences souvent « socialement-incompatibles » (rare sont les développeurs artistes ou les artistes développeurs)
- Compétences à haut degré de connaissance

Web en général

- **Problème plus global :**

« *Pourquoi a-t-on arrêté de réfléchir dans le monde du développement web ?* »

- Décalage fort entre l'informatique à l'université et en « entreprise »
- Abandon des réflexions
 - Sur le stockage des données (BD obligatoire) => *Quid des systèmes de fichiers ? Quid des données applicatives ?*
 - Sur les performances
 - Sur l'architecture générale (*Client-Serveur toujours pertinent en 2020 ?*)
 - Phénomène de mode qui l'emporte sur le reste

Web en général

- **Enjeux énormes !**

« *Pourquoi gaspiller autant de ressources ? (mémoire, électrique, temps de travail, etc.)* »

« *Pourquoi une technologie aussi basique (mise-en-forme de texte) est-elle devenue si complexe ?* »

- **De vraies questions à se poser :**

« *N'a-t-on pas poussé le web trop loin ?* »

⇒ Trouver une alternative pour faire des applications sans clients à installer et accessibles sur Internet ?

(HTML pertinent aujourd'hui pour des applications web lourde ?)

« *Que peut-on faire pour améliorer les choses ?* »

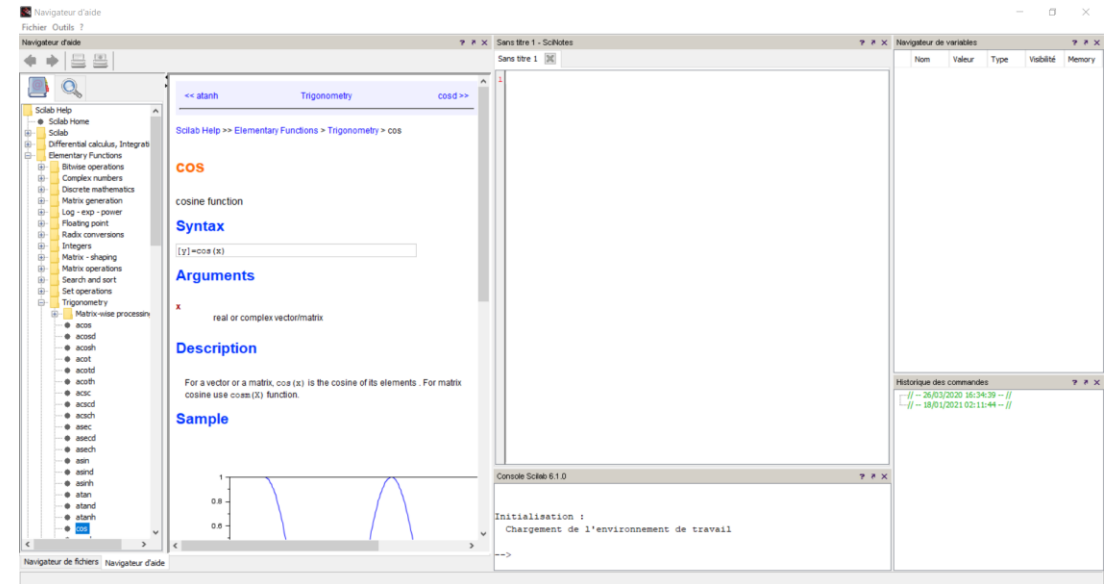
=> A défaut de changer le monde du web : il faut penser une nouvelle façon de développer en web/développer un outil pour tous

Idée Générale

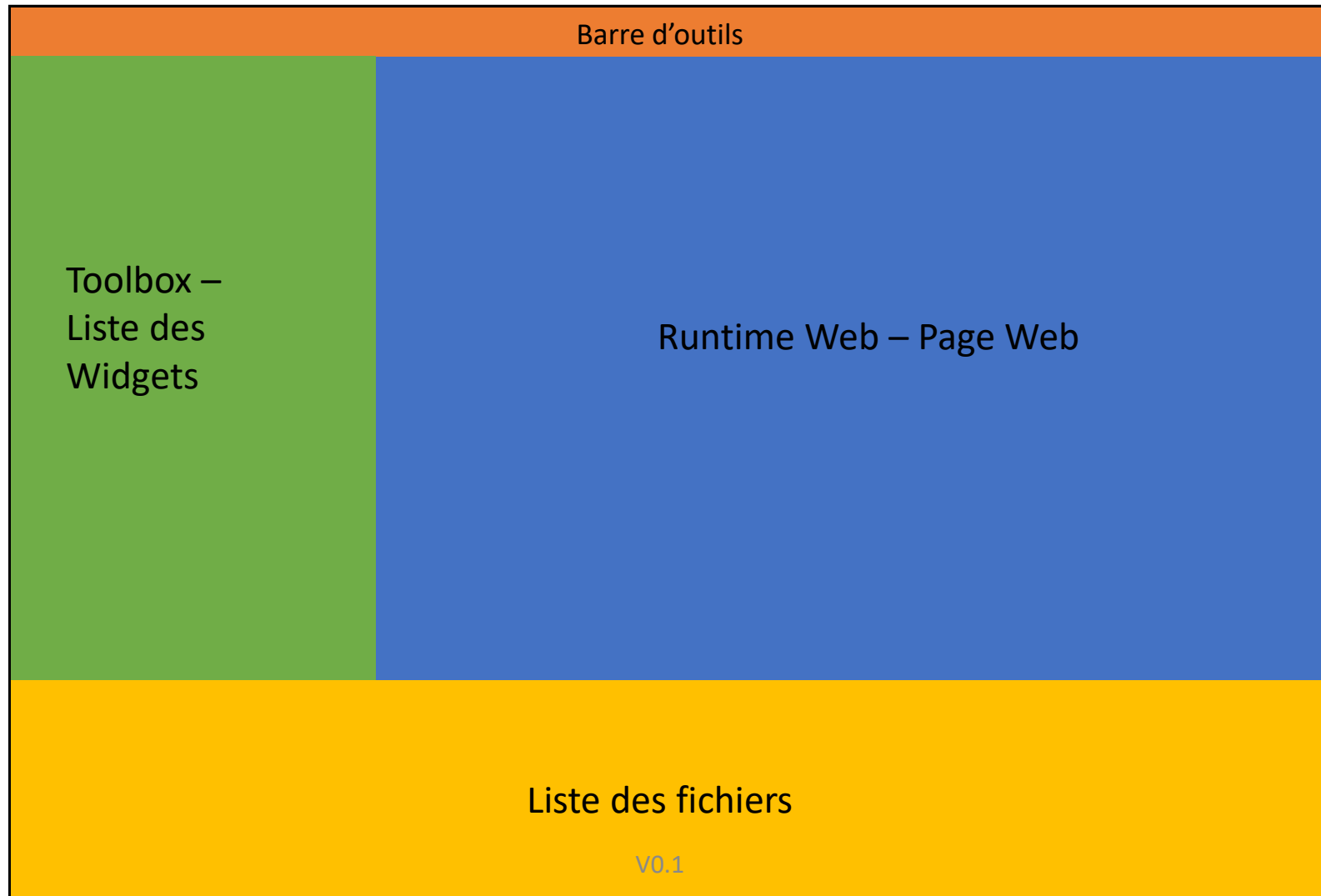
- Une façon simple et efficace de faire du web (accessible à tous ?)
- Un outil aussi bien pour les designers que les développeurs !
- Un Concepteur d'IHM/IDE/Debugger moderne (Tout en un)
- Sortir de la logique du Framework web
 - Pour des raisons de performances
 - Plus d'utilité (composant intégrés directement)
 - Retour à un HTML-CSS-Javascript
- On cherche à créer un outil pour faire ce que fait un web designer !
=> Le web designer

Interface

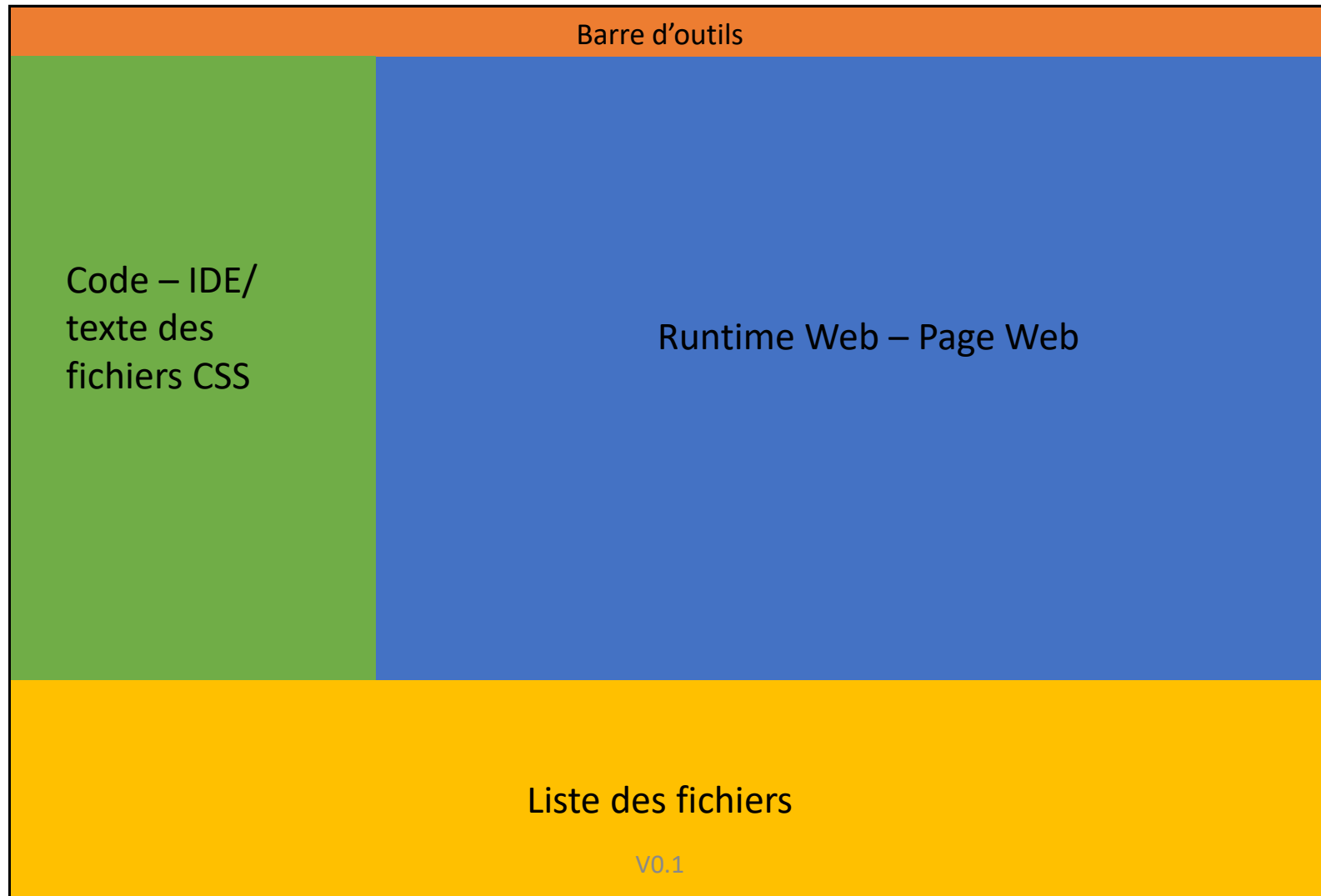
- Créer une **interface riche et flexible**
- Inspirée de l'interface Scilab
- **Enregistrement de preset/configurations ?**
- Barre d'outil qui s'adapte selon la fenêtre active ?



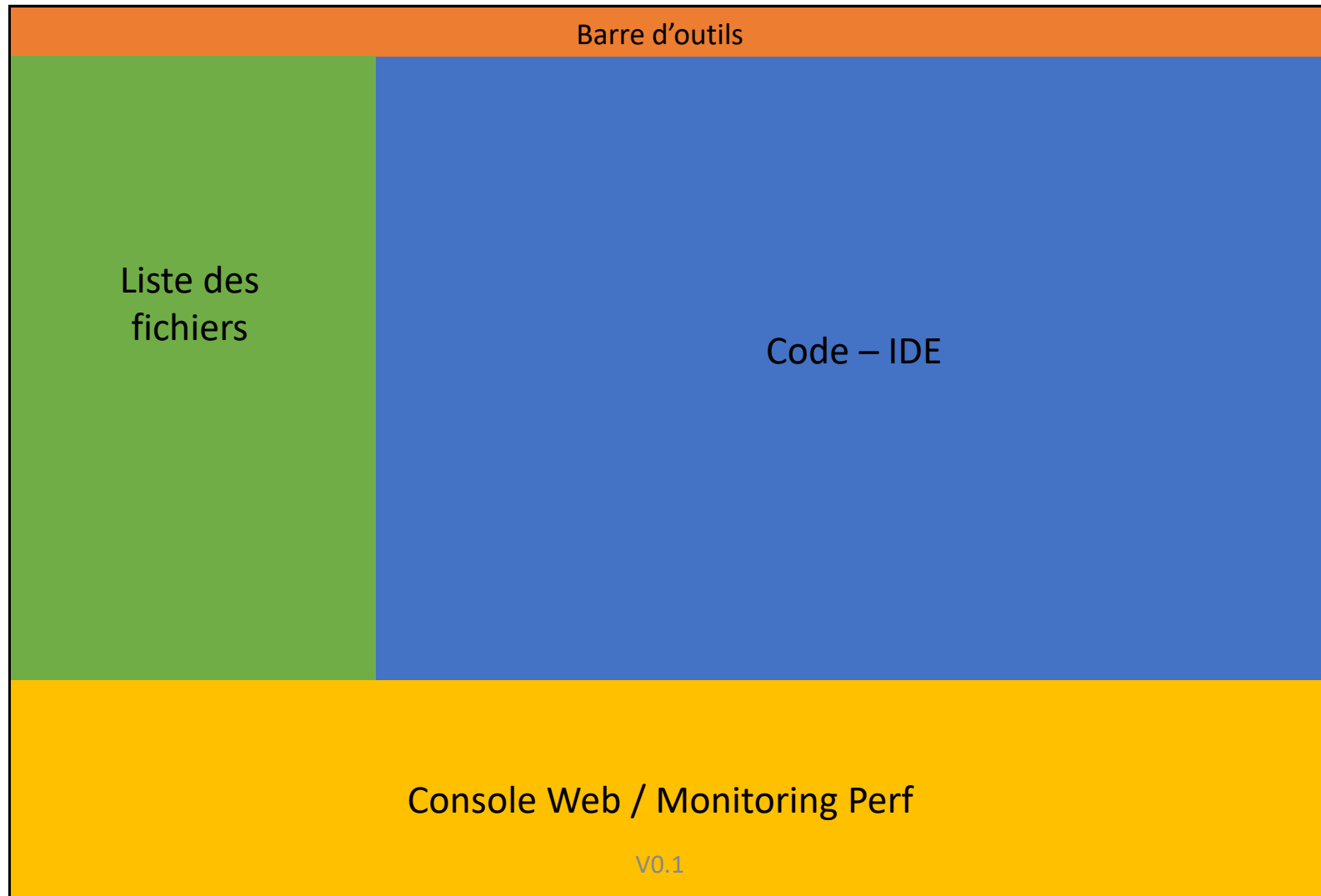
Interface - Designer



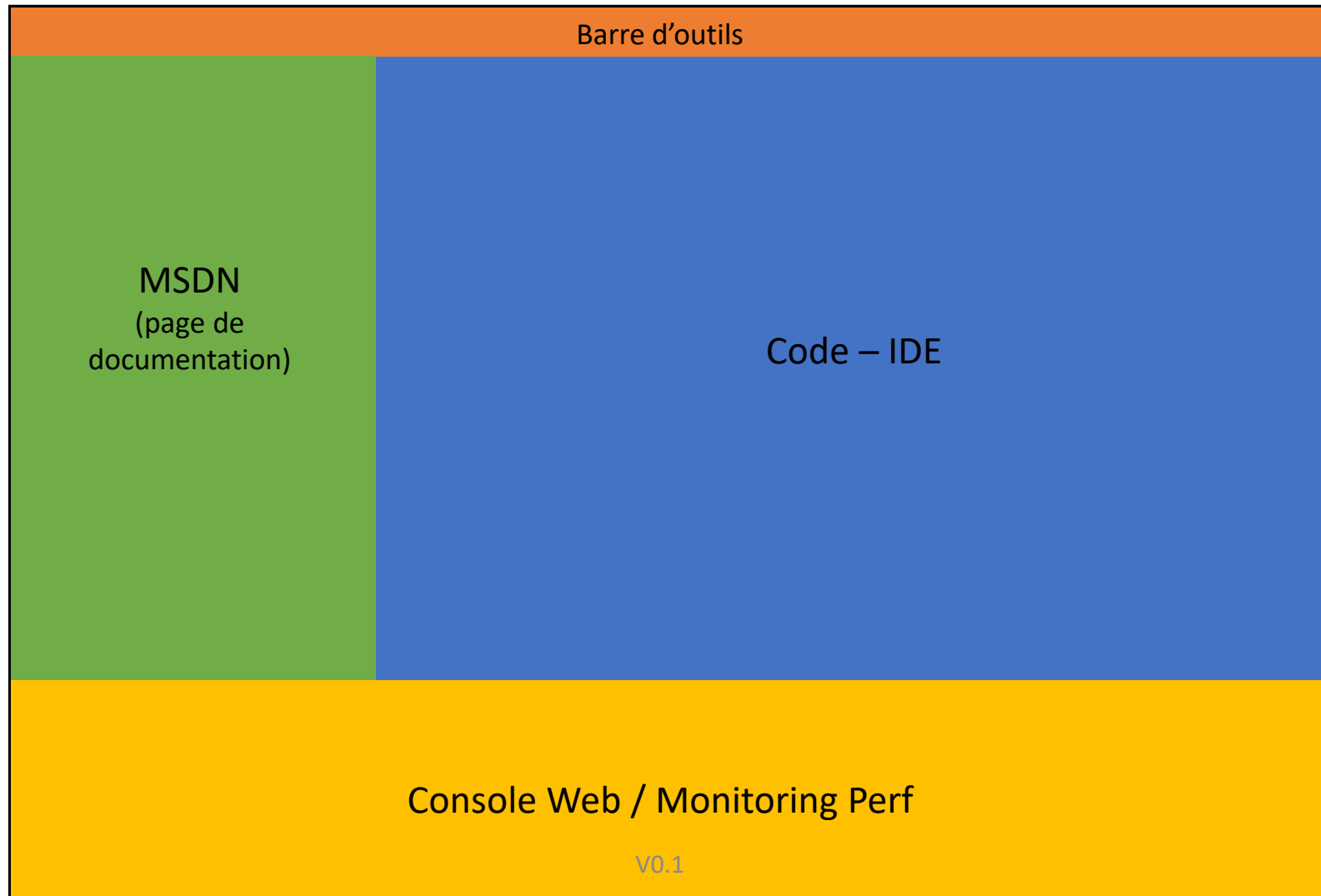
Interface – « Technical Artist »



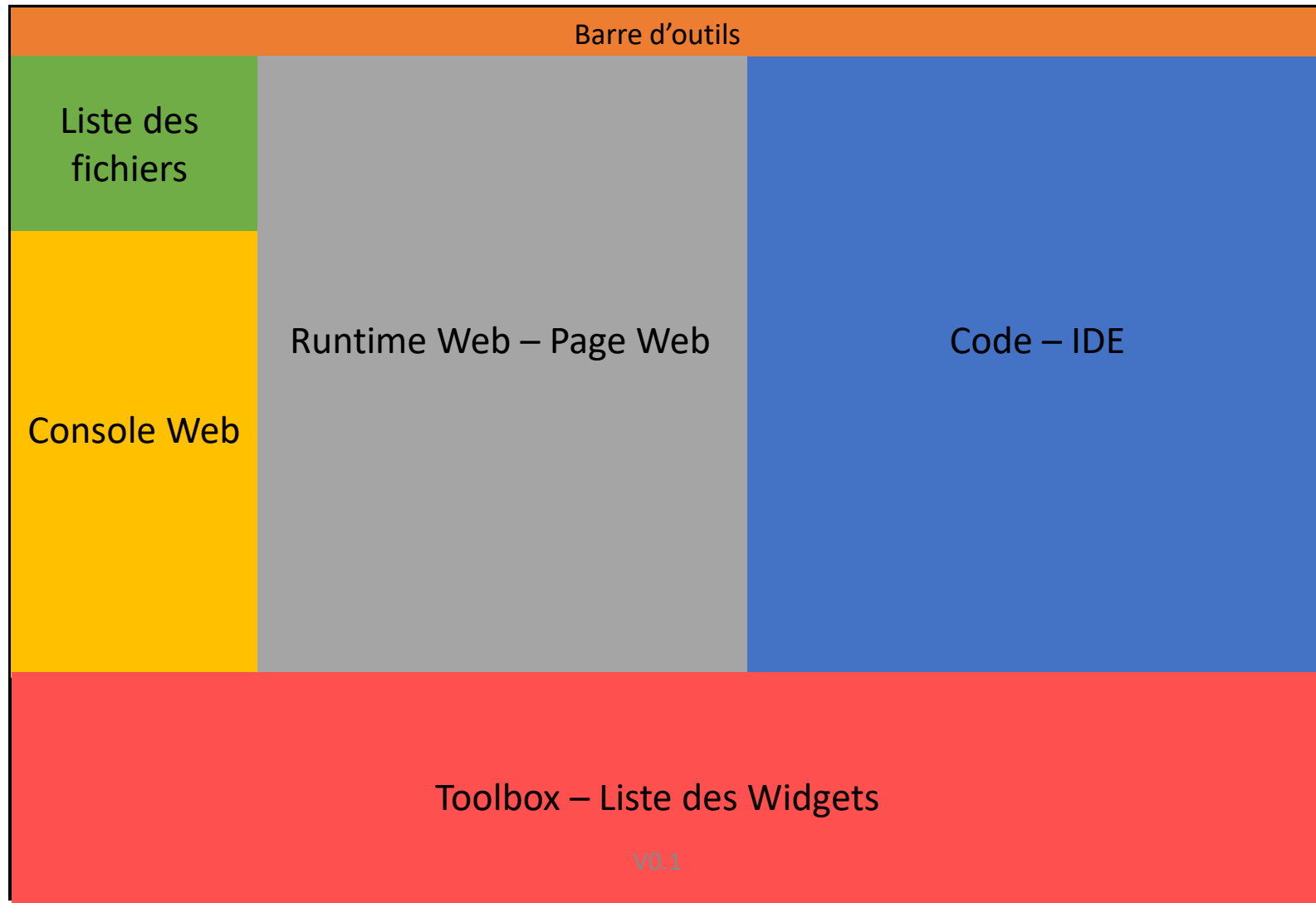
Interface – Développeur



Interface – Développeur Bis



Interface – Développeur 2




Liste des panneaux possibles

- IDE – Éditeur de Texte/code
- Runtime Web – Page Web
- Console Web – Monitoring Perf
- Explorateur de fichiers – Liste de fichiers
- Toolbox – Liste des widgets
- Panneau de config d'un widget sélectionné
- Documentation ? (Onglet Web ou intégré ?)

IDE – Éditeur de Texte/code

- Contient:
 - Coloration syntaxique
 - Système de vérification de balisage (cf. fermeture d'accolades)
 - Overview fichier
 - Raccourcis basiques
 - Analyse « sémantique » basique (cf. Intellisense sur VS)



Runtime Web – Page Web

- Lance un serveur localhost ?
- Juste rendu d'une page web ?

Console Web – Monitoring Perf

- La console de base de n'importe quel navigateur web
- Inclure la partie performance de la console ? (les mettre ensemble ou les séparer)
- Inclure la partie inspecteur de la console ? (les mettre ensemble ou les séparer)
- Principal outil de monitoring/débug ?

