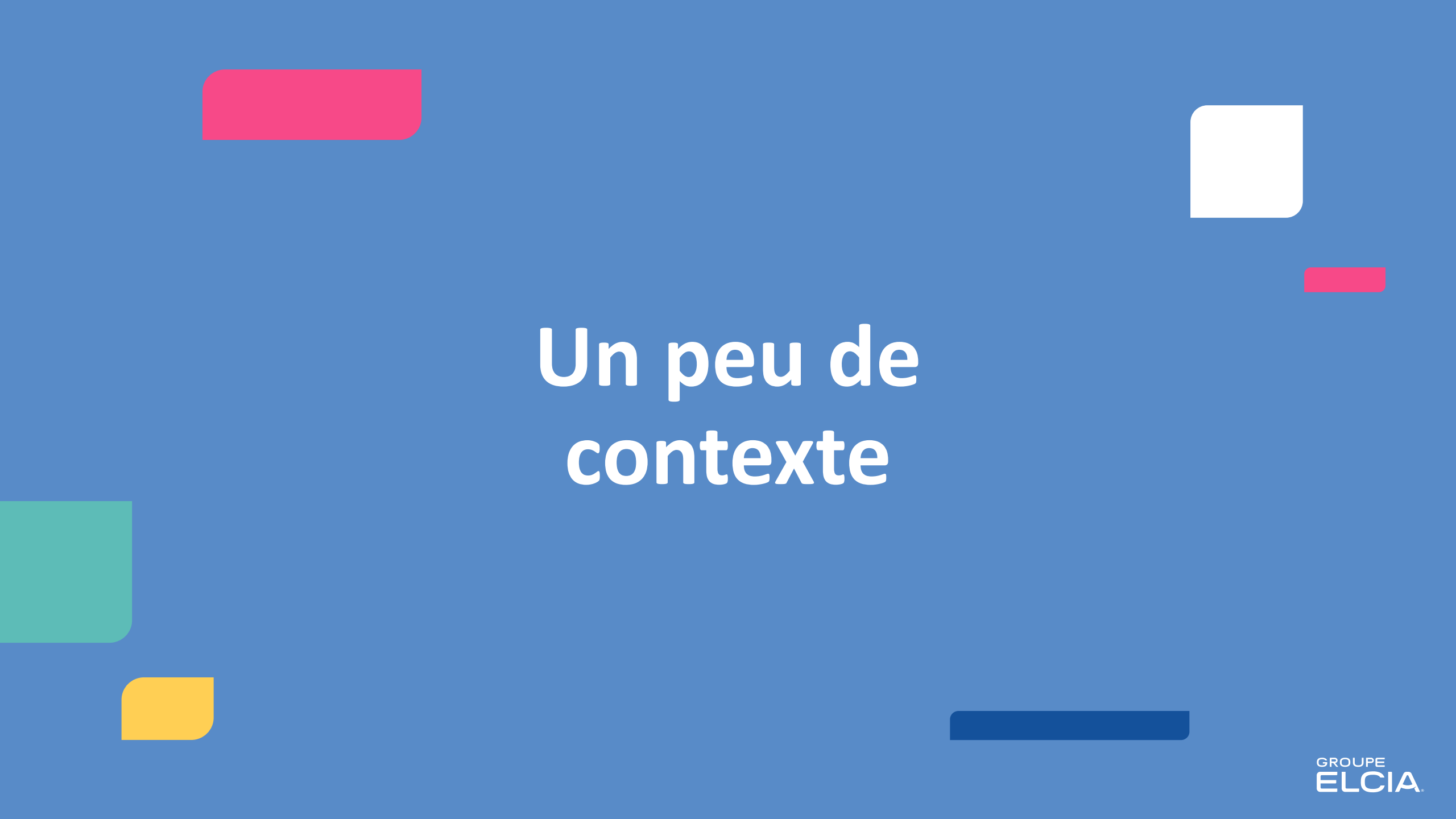




# Modernisation d'un client lourd

# Sommaire

- » Un peu de contexte
- » Que faire ?
- » Comment ça marche ?
- » Exemple de code
- » Conclusion



# Un peu de contexte

# Ce dont on a besoin

## Volumétrie

**2 000+**  
écrans générés  
via un moteur interne

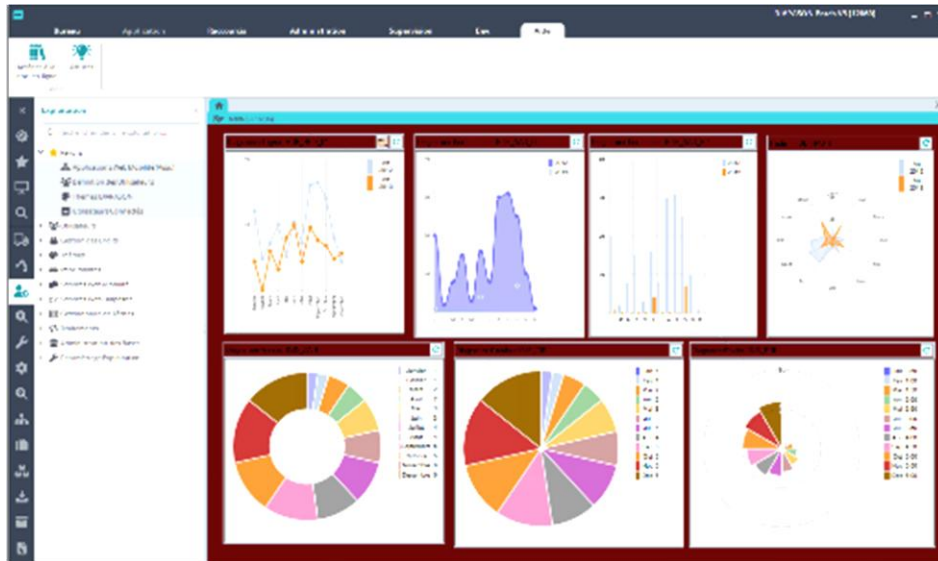
## Personnalisation

**Des dizaines  
d'écrans sur mesure  
déjà déployés chez nos  
clients**

## Complexité

**Des demandes  
de plus en plus complexes**

# Ce qu'on a



## Application .Net WinForms

Point de départ graphique du client.

## Client ABL

Le client ABL est le point d'entrée vers le serveur d'application métier. Les composants ABL sont intégrés dans le client .Net via la classe « **WindowContainer** »

## Encapsulation du composant ActiveX

Les composants ActiveX legacy sont conservés mais isolés pour éviter la perte fonctionnelle.

## UI Telerik pour modernisation

La couche Telerik Winforms (via TPF) améliore l'ergonomie avec des composants graphiques .Net avancés.

# Bilan

- » On est limité pas la techno-stack
  - » De moins en moins maintenue (Winforms et activeX)
  - » Pas (ou très peu) de Transparence
  - » Pas d'animations
  - » Grosse charge mémoire
  - » Problème de testabilité
  - » Déploiements lourds et désynchronisés



Que faire ?

# Quelles solutions possibles ?

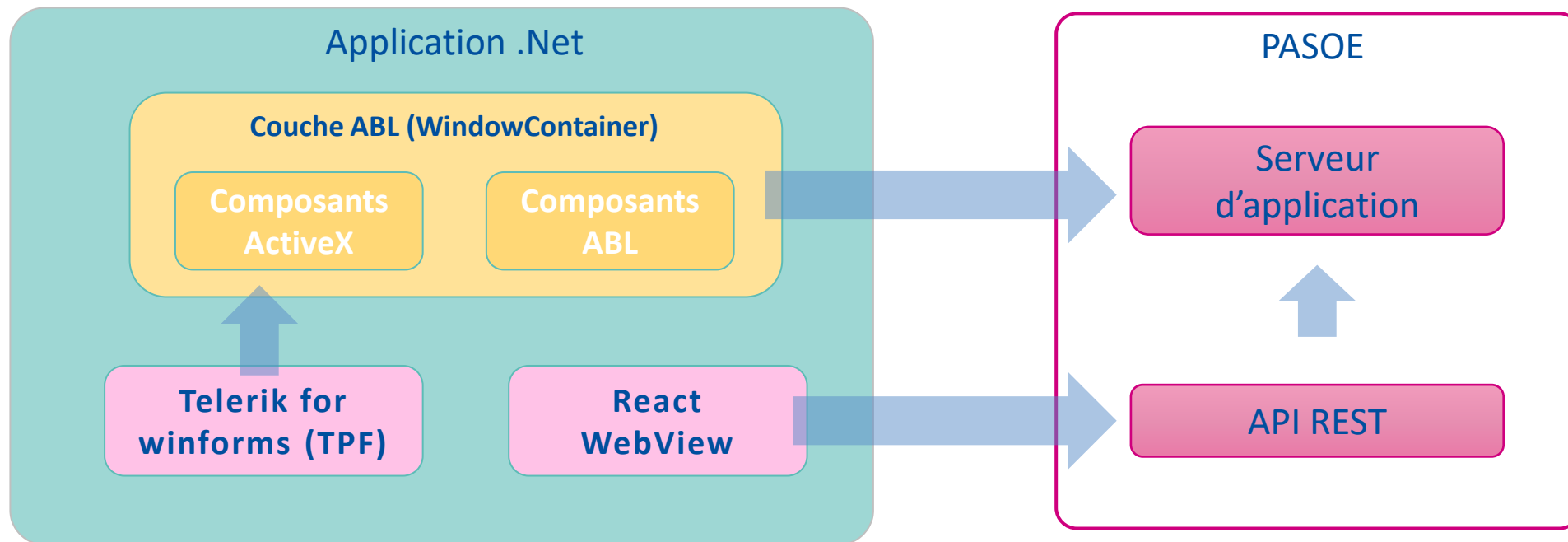
- » Faire un client en mode console (ou pas)
- » Continuer avec TPF
- » Trouver une autre techno client lourd (pourquoi pas WPF)
- » Faire un client full web (cloud ou local)

Vers une solution  
évolutive

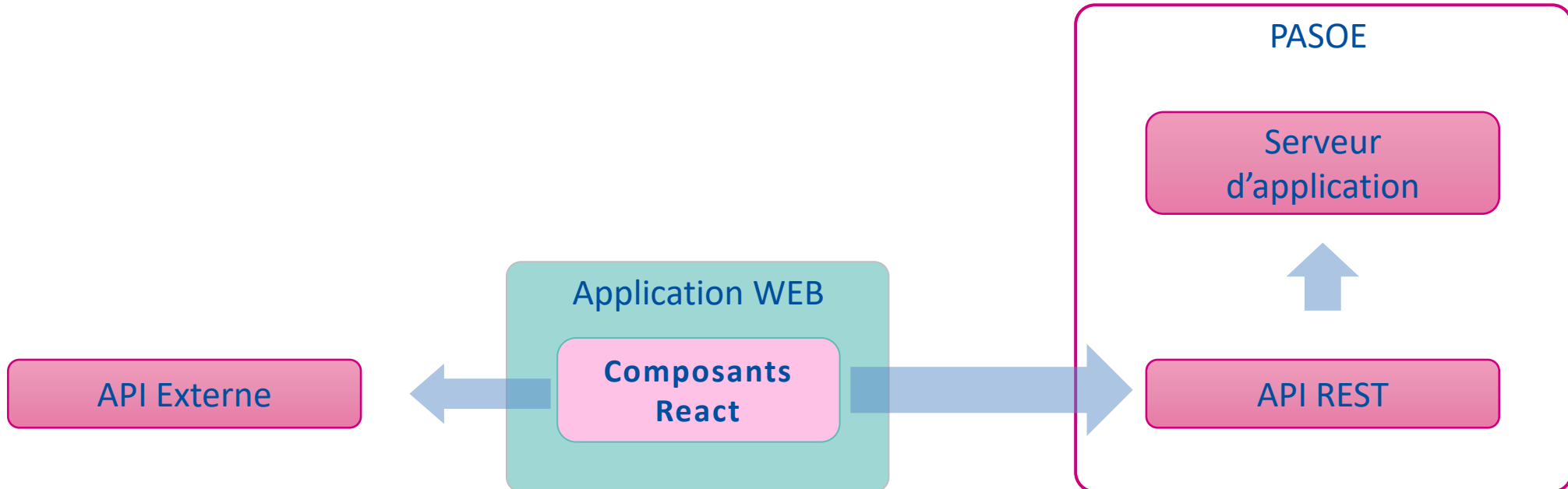
## Nos contraintes pour avancer

- » Interaction avec l'existant
- » Déploiement simple
- » Protocole d'échange réutilisable pour la suite
- » Un retour utilisateur rapide

# Intégrer des composants Web dans le client lourd



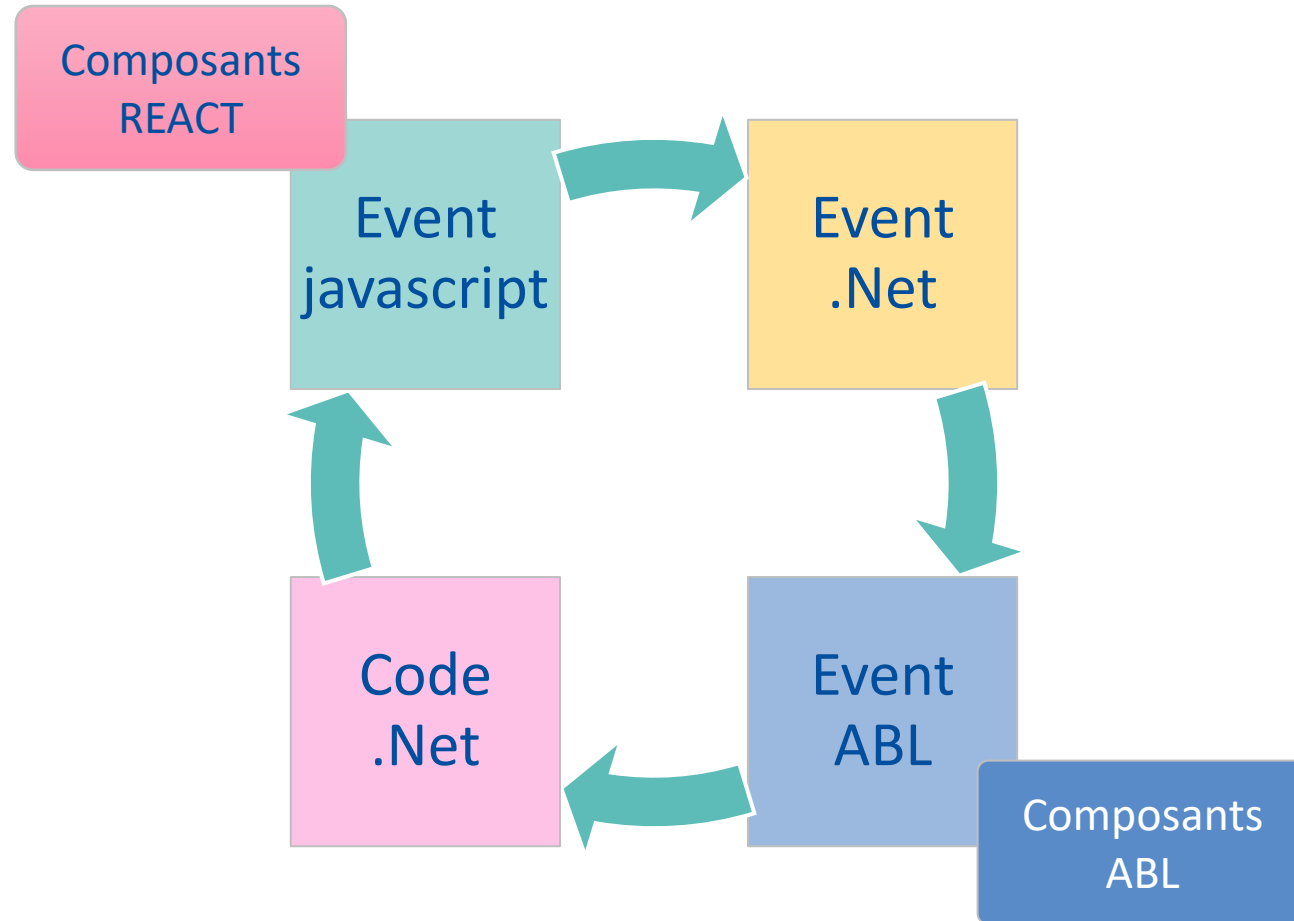
# Réutilisation des composants





# Comment ça marche

# Echange bidirectionnel

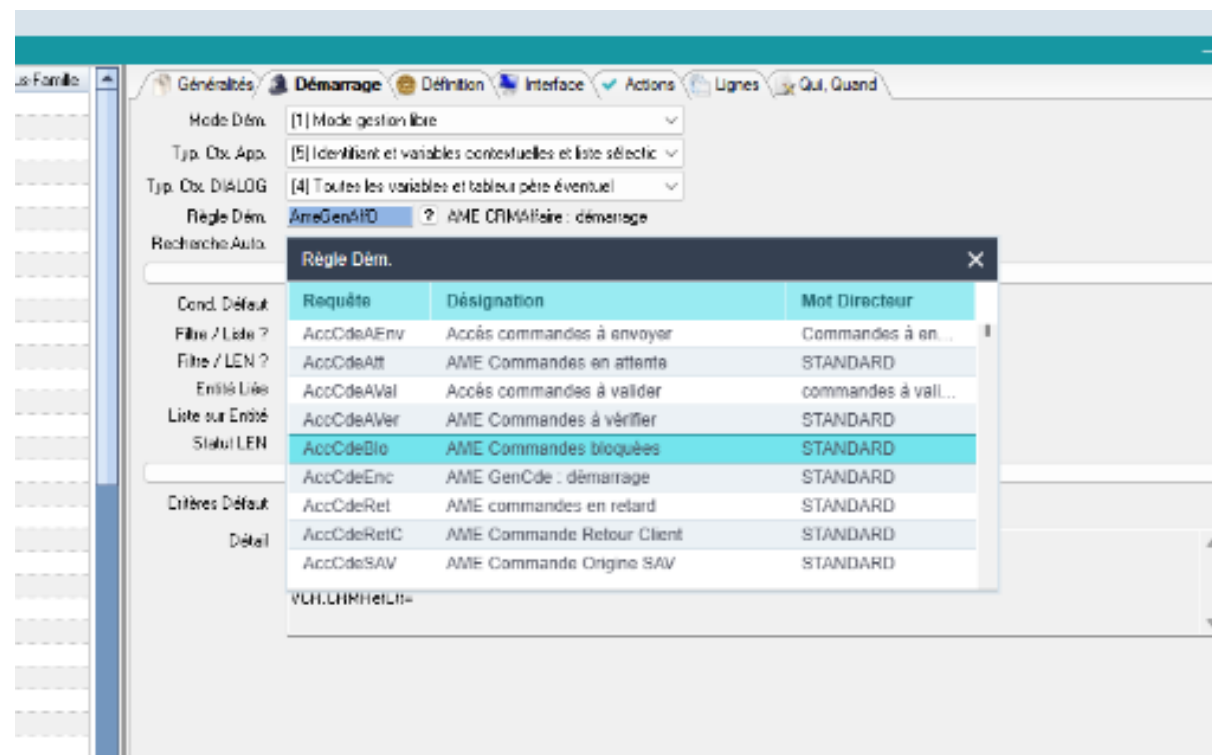


# Déploiement

- » Directement dans le PASOE
- » Simple avec un serveur web (IIS, nginx)
- » Possibilité de le faire directement dans l'application Winforms (hôte virtuel)

# Nouvelles listes d'aide

En cours de finalisation



The screenshot displays a software configuration window titled 'Règle Dém.' with a close button (X). The window is divided into a left sidebar and a main table area. The sidebar contains various configuration options, and the main area contains a table with three columns: 'Requête', 'Désignation', and 'Mot Directeur'. The table lists several request codes and their corresponding descriptions and director words.

Requête	Désignation	Mot Directeur
AccCdeAEnv	Accès commandes à envoyer	Commandes à en...
AccCdeAtt	AME Commandes en attente	STANDARD
AccCdeAVal	Accès commandes à valider	commandes à vall..
AccCdeAVer	AME Commandes à vérifier	STANDARD
AccCdeBlo	AME Commandes bloquées	STANDARD
AccCdeEnc	AME GenCde : démarrage	STANDARD
AccCdeRet	AME commandes en retard	STANDARD
AccCdeRetC	AME Commande Retour Client	STANDARD
AccCdeSAV	AME Commande Origine SAV	STANDARD

# Nouvelles listes

En cours d'étude fonctionnelle

The screenshot shows a software application window titled 'AddOne [100004]'. The main menu includes 'Bureau', 'Application', 'Rapports', 'Administration', 'Supervision', 'Dev.', and 'Aide'. The 'Application' menu is open, showing options like 'Niveau', 'Zone', 'Article', 'Vues', 'Doc', and 'Actions'. The 'Stocks' menu is also open, listing various stock management functions. The main data table is titled 'Application RSM...' and contains the following data:

Date	Niveau	Article	Section	Emplacement	Utilisateur	Movement	Type Mat.
13/11/25	50074	T00001	01-FXP	01-FXP	isa	FN	FN
13/11/25	50554	T00001	01-FAB	01-FAB	isa	EN	EN
13/11/25	50604	T00001	01-FXP	01-FXP	isa	FN	FN
13/11/25	50914	T00001	01-MAG	01-MAG	isa	EN	EN
13/11/25	50877	T00001	01-CTR	01-CTR	isa	SO	SO
13/11/25	50880	T00001	01-RETOU	01-RETOU	isa	EN	DC
13/11/25	50880	T00001	01-FNB	01-FNB	isa	SO	DE
13/11/25	50643	T00001	01-MAG	01-MAG	isa	SO	SO
13/11/25	50815	T00001	01-CIK	01-CIK	isa	EN	EN
13/11/25	50100	T00001	01-FXP	01-FXP	isa	SO	SO
13/11/25	50818	T00001	01-L20"	01-L20"	isa	EN	EN
13/11/25	50644	T00001	01-FXP	01-FXP	isa	FN	FN
13/11/25	55884	T00001	01-L20"	01-L20"	isa	EN	EN
13/11/25	50528	T00001	01-MAG	01-MAG	isa	EN	DE
13/11/25	55528	T00001	01-CTR	01-CTR	isa	SO	DC
13/11/25	50450	T00001	01-MAG	01-MAG	isa	EN	DE
13/11/25	55450	T00001	01-CTR	01-CTR	isa	SO	DC
13/11/25	50558	T00001	01-CIK	01-CIK	isa	SO	SO
13/11/25	50228	T00001	01-CTR	01-CTR	isa	SO	SO
08/11/25	51421	T00001	01-FAB	01-FAB	isa	EN	EN

# Monitoring

En cours d'étude fonctionnelle





# Exemple de code

# Message vers Progress

## Coté React

```
const message = JSON.stringify(dataForMessage)
if (window.chrome !== undefined && window.chrome.webview !== undefined)
  window.chrome.webview.postMessage(message);
else
  console.log(message);
```

## Coté .Net

```
...
Browser.WebMessageReceived += Browser_WebMessageReceived;
...

protected virtual void Browser_WebMessageReceived(object sender, Microsoft.Web.WebView2.Core.CoreWebView2WebMessageReceivedEventArgs e)
{
  var receivedMessage = e.TryGetWebMessageAsString();
  if (receivedMessage != null)
    WebMessageReceived?.Invoke(this, new WebMessageReceivedEventArgs() { ReceivedMessage = receivedMessage });
}
```

## Coté Progress

```
webApplication.WebMessageReceived.Subscribe(WebMessageReceivedHandler).
```

# Message vers React

## Coté .Net

```
// Envoyer un message vers React
public void SendMessageToReact(object payload)
{
    string json = JsonSerializer.Serialize(payload);
    webView21.CoreWebView2.PostWebMessageAsJson(json);
}
```

## Coté React

```
useEffect(() => {
    return onMessage("update", (payload) => {
        console.log("Reçu de .NET :", payload);
    });
}, []);
```

# Conclusion

## Conclusion

- » Une solution WEB en React
- » Un déploiement simple
- » Ouverture technique et ergonomique
- » Solution adaptée pour un client lourd, Web ou mixte



Questions ?

GROUPE  
ELCIA®

*Merci*

C'est beau .... hein

