

SPECIFICATIONS FONCTIONNELLES GENERALES

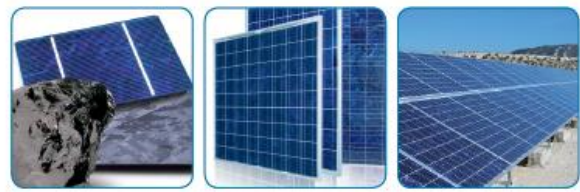
ETUDE TECHNIQUE POUR CDSN

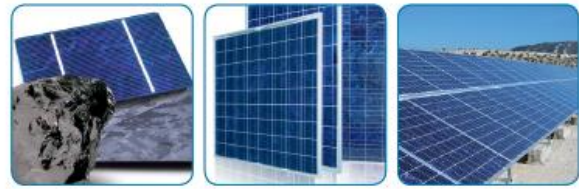
SFG – ETUDE TECHNIQUE CDSN – 2018 – 01

INDICE	N° PAGES	DESCRIPTION DE L'EVOLUTION	DATE
V1.0	Toutes	Création du document	12/02/2018

Destinataires :

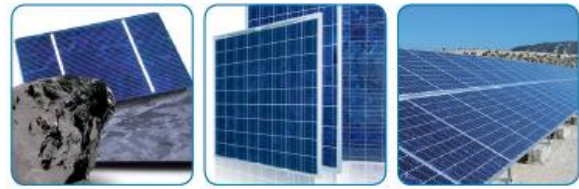
SOCIETE	NOM ET PRENOM	SERVICE
		SI
		SI
		SI
		SI
		SI





SOMMAIRE

1. OBJECTIF DE L'ETUDE	4
2. CONTEXTE OPERATIONNEL & TECHNIQUE / TRAVAUX DE REFERENCE.....	5
2.1 Contexte Opérationnel	5
2.2 Travaux de Référence	6
3. RECENSEMENT DES SOLUTIONS POSSIBLES	8
4. EVALUATION DES DIFFERENTES SOLUTIONS	13
-	26
5. RESULTATS DE L'ETUDE.....	26
5.1 Synthèse des scoring des solutions	26
5.2 Annexes	28



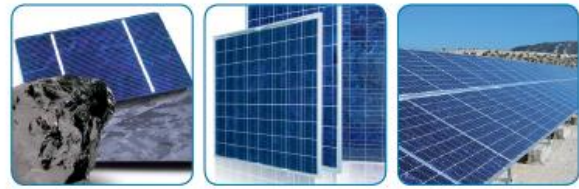
1. OBJECTIF DE L'ETUDE

L'application CDSN est amenée à évoluer pour cadrer avec les nouvelles exigences de la nouvelle entité PhotoWatt Crystal Advanced.

Pour anticiper cette évolution, il est nécessaire d'étudier les solutions techniques qui seraient les plus appropriées en tant que nouveau support de réécriture de cette application.

Le document a donc pour but de :

- Recenser les différentes solutions actuellement disponibles sur le marché,
- Décrire brièvement l'architecture de ces solutions techniques,
- Evaluer et donner un scoring de pertinence à chacune de ces solutions à l'aune des contraintes globales de l'application et du contexte PhotoWatt



2. CONTEXTE OPERATIONNEL & TECHNIQUE / TRAVAUX DE REFERENCE

2.1 Contexte Opérationnel

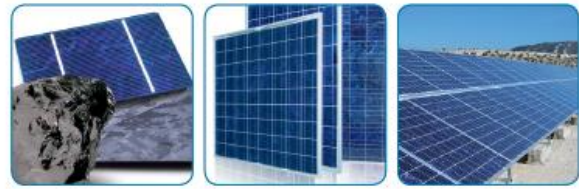
Ce cahier des charges s'inscrit dans le cadre du projet d'évolution de l'application CDSN 2019.

Version du logiciel actuel = CDSN 4.23.2

Rappel des exigences sommaires et non exhaustives de l'application :

- Etre déployé sur plusieurs postes utilisateurs de type PC/ Laptop,
- Communiquer avec des bases de données,
- Proposer des écrans de saisie par le biais de formulaires,
- Proposer des états récapitulatifs et des synthèses de données,
- Pouvoir gérer des imprimantes pour l'impression d'étiquettes d'emballage,
- Etre potentiellement déployé sur des périphériques mobiles de type tablette,

Le budget de développement souhaité pour l'évolution de cette application serait de 3 mois.

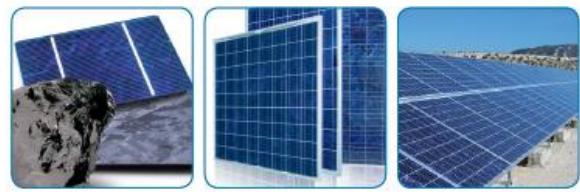


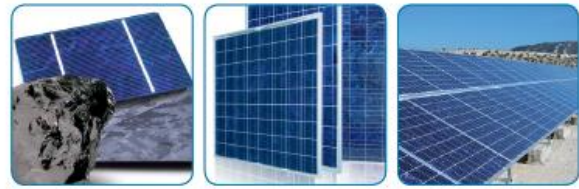
2.2 Travaux de Référence

Toutes les spécifications existantes de l'application de traçabilité CDSN.

Les documents de synthèse de l'étude technique :

« Choix techniques – CDSN.xlsx »





3. RECENSEMENT DES SOLUTIONS POSSIBLES

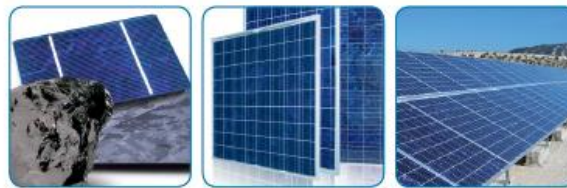
Nous avons recensé environ 13 solutions techniques possibles pour cette application en considérant :

- Le type d'architecture,
- L'environnement de développement,
- Les Frameworks disponibles,
- L'éditeur potentiel de la solution,
- Les langages utilisés,

Nous n'avons considéré que les technologies récentes et compatibles avec l'infrastructure et l'environnement de PhotoWatt.

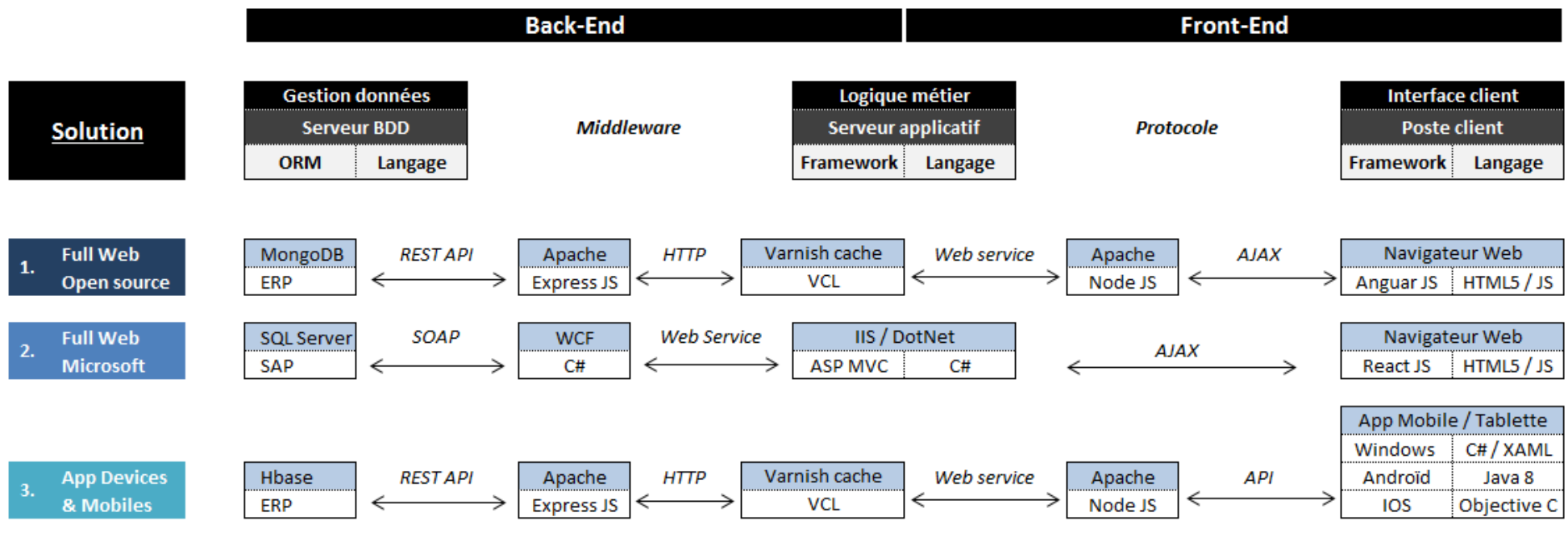
Ces solutions ont été classées par type d'architecture qui se déclinent selon la liste suivante :

- Architecture micro-services,
- Architecture distribuée n-tiers,
- Architecture hybride,
- Architecture client lourd,
- Architecture propriétaire (ERP),

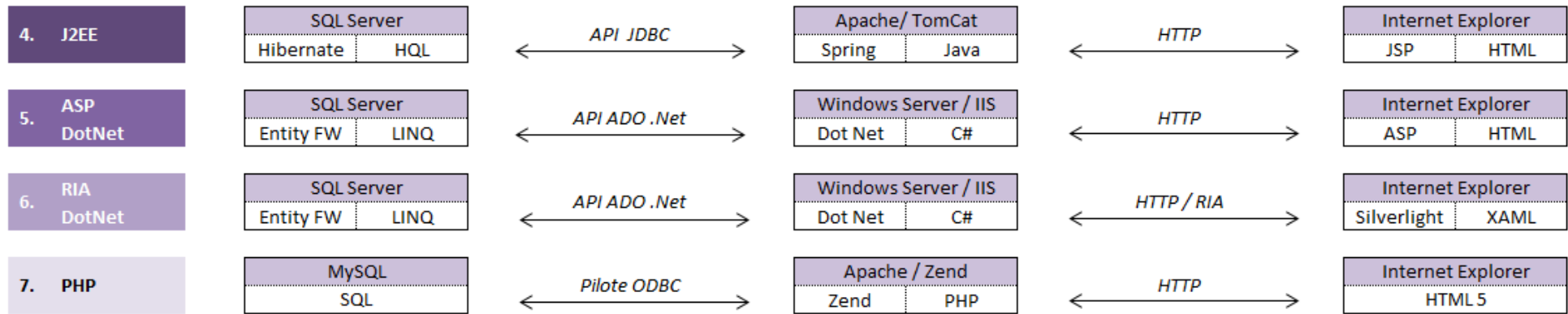


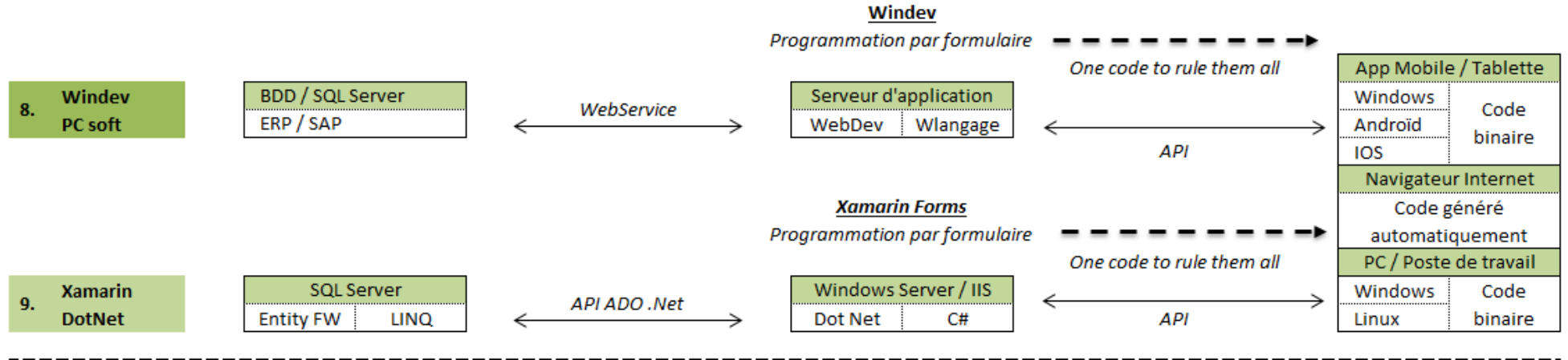
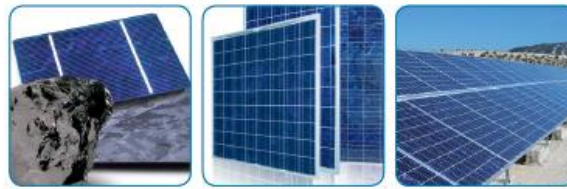
ARCHITECTURE DES SOLUTIONS POSSIBLES POUR LE FUTUR CDSN - VERSION 2019

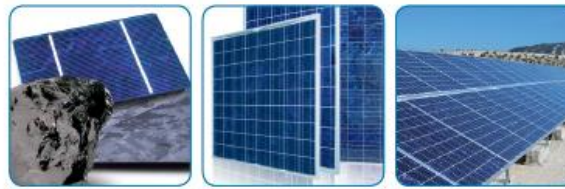
Architecture micro services



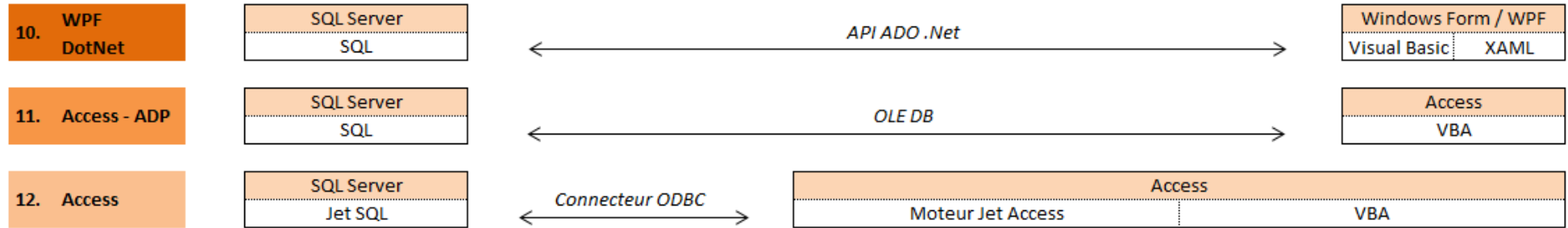
Courbe d'apprentissage



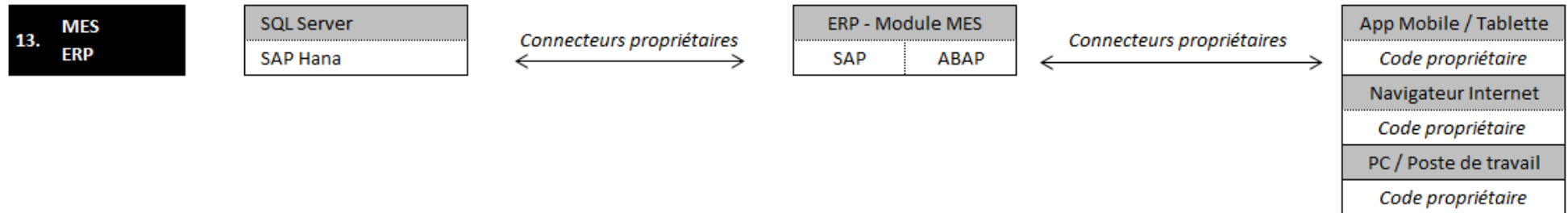




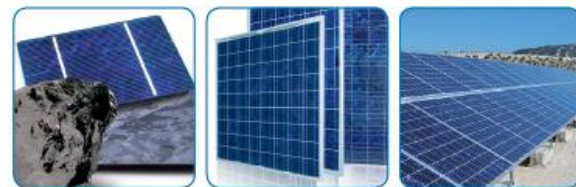
Architecture client lourd



Architecture ERP



entissage faible



4. EVALUATION DES DIFFERENTES SOLUTIONS

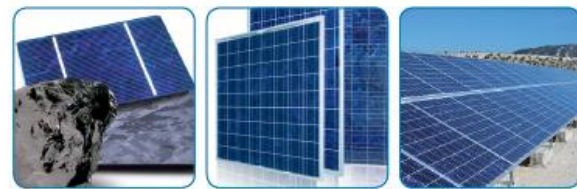
1. Solution Full Web Open Source

Type Architecture	Architecture micro serveurs
Interface utilisateur	Navigateur Web
Langage(s)	HTML5 - Javascript
Opérationnel - coefficient 1 <ul style="list-style-type: none"> • Coût par utilisateur : ● • Courbe d'apprentissage : ● • Souplesse de déploiement : ● 	
Performance - coefficient 1 <ul style="list-style-type: none"> • Résistance aux montées en charge potentielles : ● • Délais et rapidité de nouveaux développements : ● • Richesse de l'expérience utilisateur : ● 	
Adaptation au contexte PhotoWatt - coefficient 1 <ul style="list-style-type: none"> • Fonctionnement avec équipe IT réduite : ● • Possibilité de développement en mode silo : ● • Pertinence vis-à-vis de l'infrastructure existante : ● 	
Adaptation au contexte CDSN - coefficient 1 <ul style="list-style-type: none"> • Dialogue avec autres composants (imprimantes...) : ● • Compatibilité avec utilisateur en atelier industriel : ● • Risque de rejets utilisateurs (aversion au changement) : ● 	
Forces <ul style="list-style-type: none"> • Les temps de développement d'un écran sont très courts grâce à une très grande factorisation des composants génériques, • Un système de binding asynchrone efficace offre des temps de traitement et d'affichage des pages Web particulièrement rapides, tout en sollicitant au minimum les bases de données, 	Faiblesses <ul style="list-style-type: none"> • L'apprentissage du framework Angular JS est particulièrement complexe et demande une bonne connaissance du Javascript, • La solution étant actuellement très demandée, le taux journalier des consultants est plus élevé que pour d'autres technologies,
Opportunités <ul style="list-style-type: none"> • Le Framework Angular JS est devenu le standard open source des applications Web développées aujourd'hui (plus du 1/3 des projets Web d'importance depuis 2013), • Bien qu'open source, le framework est développé par les équipes de Google, ce qui assure une évolution et une amélioration constante sur les années à venir, 	Obstacles <ul style="list-style-type: none"> • Les applications purement Web sont avant tout destinées à des environnements complètement dématérialisés et les communications avec des périphériques clients lourds peuvent s'avérer parfois complexes (imprimantes, automates), • Les interactions avec les composants systèmes sont parfois plus difficiles qu'avec un client lourd (accès API Win32),

Score total : 14 sur 36 - Pourcentage de scoring : 38,89 %



Conclusion : L'application Full Web permet de mettre à disposition l'application sur tout type d'écran (ordinateur, tablettes) très facilement, mais ce n'est pas forcément le choix le plus approprié pour le contexte industriel de CDSN (impression d'étiquettes...)



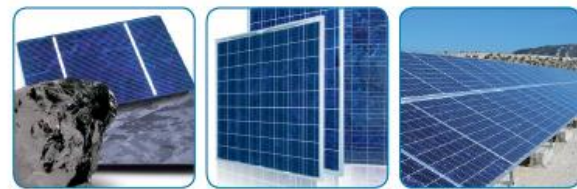
2. Solution Full Web Microsoft

Type Architecture	Architecture micro serveurs
Interface utilisateur	Navigateur Web
Langage(s)	HTML5 - Javascript - C#
<p>Opérationnel - coefficient 1</p> <ul style="list-style-type: none"> • Coût par utilisateur : ● • Courbe d'apprentissage : ● • Souplesse de déploiement : ● <p>Performance - coefficient 1</p> <ul style="list-style-type: none"> • Résistance aux montées en charge potentielles : ● • Délais et rapidité de nouveaux développements : ● • Richesse de l'expérience utilisateur : ● 	
<p>Adaptation au contexte PhotoWatt - coefficient 1</p> <ul style="list-style-type: none"> • Fonctionnement avec équipe IT réduite : ● • Possibilité de développement en mode silo : ● • Pertinence vis-à-vis de l'infrastructure existante : ● <p>Adaptation au contexte CDSN - coefficient 1</p> <ul style="list-style-type: none"> • Dialogue avec autres composants (imprimantes...) : ● • Compatibilité avec utilisateur en atelier industriel : ● • Risque de rejets utilisateurs (aversion au changement) : ● 	
<p>Forces</p> <ul style="list-style-type: none"> • Le mode responsive design permet un affichage stable et homogène autant sur PC que périphériques mobiles, • Un système de binding asynchrone efficace offre des temps de traitement et d'affichage des pages Web particulièrement rapides, tout en sollicitant au minimum les bases de données, 	<p>Faiblesses</p> <ul style="list-style-type: none"> • Comme pour Angular JS, l'apprentissage de React JS peut s'avérer complexe et demande une bonne connaissance du Javascript, • Il est rare que le même consultant soit Full stack et maîtrise toute la chaîne de développement en raison des multiples technologies à connaître, d'où un prix plancher du projet potentiellement plus élevé,
<p>Opportunités</p> <ul style="list-style-type: none"> • Le Framework React JS s'inscrit dans l'éco-système DotNet qui est l'un des plus dynamique du marché, • La solution étant full Microsoft, le support et l'accompagnement éditeur est possible, 	<p>Obstacles</p> <ul style="list-style-type: none"> • Les applications purement Web sont avant tout destinées à des environnements complètement dématérialisés et les communications avec des périphériques clients lourds peuvent s'avérer parfois complexes (imprimantes, automates), • Les interactions avec les composants systèmes sont parfois plus difficiles qu'avec un client lourd (accès API Win32),

Score total : 14 sur 36 - Pourcentage de scoring : 38,89 %



Conclusion : L'application Full Web permet de mettre à disposition l'application sur tout type d'écran (ordinateur, tablettes) très facilement, mais ce n'est pas forcément le choix le plus approprié pour le contexte industriel de CDSN (impression d'étiquettes...)



3. Solution App & devices mobiles

Type Architecture	Architecture micro serveurs
Interface utilisateur	Navigateur Web
Langage(s)	C# - XAML - Java - Objective C

Opérationnel - coefficient 1

- Coût par utilisateur : ●
- Courbe d'apprentissage : ●
- Souplesse de déploiement : ●

Performance - coefficient 1

- Résistance aux montées en charge potentielles : ●
- Délais et rapidité de nouveaux développements : ●
- Richesse de l'expérience utilisateur : ●

Adaptation au contexte PhotoWatt - coefficient 1

- Fonctionnement avec équipe IT réduite : ●
- Possibilité de développement en mode silo : ●
- Pertinence vis-à-vis de l'infrastructure existante : ●

Adaptation au contexte CDSN - coefficient 1

- Dialogue avec autres composants (imprimantes...) : ●
- Compatibilité avec utilisateur en atelier industriel : ●
- Risque de rejets utilisateurs (aversion au changement) : ●

Forces

- La partie Front-End étant spécifique à chaque périphérique, il est possible de personnaliser l'application pour chaque type de support (les besoins pour un PC n'étant pas les mêmes que pour une consultation sur tablette),
- Sur les périphériques mobiles, les applications en mode Web utilisent des services Worker qui ralentissent les performances, quand les applications développées en mode natif sont beaucoup plus puissantes et rapides,

Opportunités

- Les écrans utilisateur étant développés en mode client lourd, les interactions avec les composants système (impression, dialogue automate) sont beaucoup plus faciles et aisés,
- On peut se contenter de ne développer que pour le type de périphérique que l'on envisage de cibler, au lieu d'avoir recours à des frameworks full web, qui permettent de l'affichage sur tout type de support au prix d'une importante complexité,

Faiblesses

- Les coûts de développement sont forcément plus élevés car il faut développer une application pour chaque type de support,
- Aucun ingénieur n'est spécialisé sur toutes les plateformes mobiles, donc généralement ce type de projet nécessite plusieurs développeurs Front End,

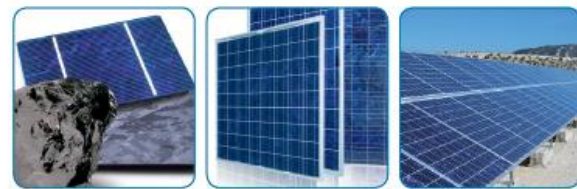
Obstacles

- Comme pour les coûts de développement, les coûts de maintenance sont démultipliés, car pour chaque modification sur un écran utilisateur nécessite un développement sur chaque application,
- Les périphériques mobiles ne sont pas forcément adaptés à la culture métallurgiste de l'atelier Wafers, encore peu informatisé et dont l'appétence en nouvelles technologies serait déjà à tester en amont,

Score total : 16 sur 36 - Pourcentage de scoring : 44,44 %



Conclusion : L'application multi périphérique est plus approprié pour le contexte industriel de CDSN (impression d'étiquettes...) que la version Full Web mais elle reste onéreuse et peut être ambitieuse pour l'environnement PhotoWatt...



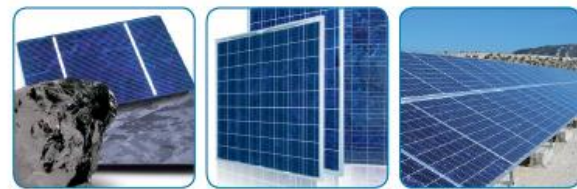
4. Solution J2EE

Type Architecture	Architecture répartie - n tiers
Interface utilisateur	Client Web
Langage(s)	Java - HQL - JSP - HTML - CSS
<p>Opérationnel - coefficient 1</p> <ul style="list-style-type: none"> • Coût par utilisateur : ● • Courbe d'apprentissage : ● • Souplesse de déploiement : ● <p>Performance - coefficient 1</p> <ul style="list-style-type: none"> • Résistance aux montées en charge potentielles : ● • Délais et rapidité de nouveaux développements : ● • Richesse de l'expérience utilisateur : ● 	
<p>Adaptation au contexte PhotoWatt - coefficient 1</p> <ul style="list-style-type: none"> • Fonctionnement avec équipe IT réduite : ● • Possibilité de développement en mode silo : ● • Pertinence vis-à-vis de l'infrastructure existante : ● <p>Adaptation au contexte CDSN - coefficient 1</p> <ul style="list-style-type: none"> • Dialogue avec autres composants (imprimantes...) : ● • Compatibilité avec utilisateur en atelier industriel : ● • Risque de rejets utilisateurs (aversion au changement) : ● 	
<p>Forces</p> <ul style="list-style-type: none"> • L'environnement J2EE est très complet et très abouti et permet la programmation de projets d'envergure, • La plate forme est en constante amélioration grâce à une communauté Open source dynamique à l'origine de nombreux Frameworks (Struts, Spring, Hibernate), 	<p>Faiblesses</p> <ul style="list-style-type: none"> • La courbe d'apprentissage est très élevée en raison du nombre élevé d'API à maîtriser (l'IDE Eclipse est également difficile), • Les frameworks sont puissants mais parfois d'une utilisation ardue et la maintenance des serveurs applicatifs est complexe,
<p>Opportunités</p> <ul style="list-style-type: none"> • L'environnement repose sur un système d'API régulièrement mis à jour ce qui le rend particulièrement modulable, • Les framework de type JUnit permettent une complète automatisation des tests unitaires, 	<p>Obstacles</p> <ul style="list-style-type: none"> • La solution J2EE n'est pas adaptée pour une infrastructure sous environnement Microsoft, • Cette plateforme est destinée avant tout aux gros projets et aux grands comptes,

Score total : 15 sur 72 - Pourcentage de scoring : 41,67 %



Conclusion : Un projet J2EE nécessite un investissement temps important et une équipe développement étoffé, ce qui offre peu d'espoir de retour sur investissement pour une PME.



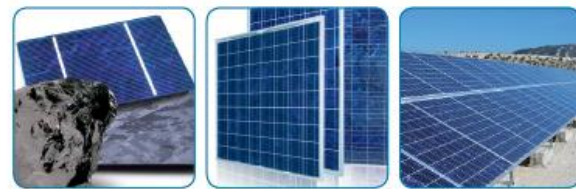
5. Solution ASP DotNet

Type Architecture	Architecture répartie - n tiers
Interface utilisateur	Client Web
Langage(s)	C# - LINK - ASP - HTML - CSS
<p>Opérationnel - coefficient 1</p> <ul style="list-style-type: none"> • Coût par utilisateur : ● • Courbe d'apprentissage : ● • Souplesse de déploiement : ● <p>Performance - coefficient 1</p> <ul style="list-style-type: none"> • Résistance aux montées en charge potentielles : ● • Délais et rapidité de nouveaux développements : ● • Richesse de l'expérience utilisateur : ● 	
<p>Adaptation au contexte PhotoWatt - coefficient 1</p> <ul style="list-style-type: none"> • Fonctionnement avec équipe IT réduite : ● • Possibilité de développement en mode silo : ● • Pertinence vis-à-vis de l'infrastructure existante : ● <p>Adaptation au contexte CDSN - coefficient 1</p> <ul style="list-style-type: none"> • Dialogue avec autres composants (imprimantes...) : ● • Compatibilité avec utilisateur en atelier industriel : ● • Risque de rejets utilisateurs (aversion au changement) : ● 	
<p>Forces</p> <ul style="list-style-type: none"> • Le mode client Web permet de centraliser les développements sur serveur, ce qui facilite déploiement, maintenance, accès aux données et gestion des utilisateurs, • Les interfaces Web offrent ergonomie et design de qualité, • Génération automatique de code et réutilisabilité de composants permettent de gros gains de temps de développement, 	<p>Faiblesses</p> <ul style="list-style-type: none"> • Bien qu'offrant la facilité d'utilisation des produits Microsoft, la puissance de la plate forme Visual Studio se paie par une courbe d'apprentissage malgré tout élevée, • Comme tout framework, il y a un phénomène d'échelle et sur de petits projets, les délais de développement ne rivalisent pas avec les solutions de programmation assistée (WinDev),
<p>Opportunités</p> <ul style="list-style-type: none"> • Le Framework WCF permet de très facilement créer des Web services au format SOAP, ce qui offre de nombreuses opportunités d'architectures applicatives, • Le Framework Entity permet de normaliser l'accès aux données et de le découpler de la logique applicative, ce qui facilite les évolutions et la maintenabilité du code, 	<p>Obstacles</p> <ul style="list-style-type: none"> • La technologie ASP commence à être dépassée et n'est plus adaptée aux nouveaux environnements mobiles (tablettes...), • Niveau infrastructure, la maîtrise des serveurs applicatifs Web IIS est nécessaire pour optimiser les temps de latence, • Des compétences en Web design et programmation HTML restent nécessaires pour la création des écrans de l'application (l'alternative RIA Silverlight offre les meilleurs rendus mais nécessite un plug-in IE problématique avec les clients légers),

Score total : 19 sur 72 - Pourcentage de scoring : 52,78 %



Conclusion : Sur les petits projets, les délais à court terme ne sont pas forcément compétitifs, mais à moyen terme des gains de temps significatifs sur les développements d'évolutions et de correctifs sont envisageables. Attention toutefois à l'obsolescence de la technologie ASP, qui ne semble pas avoir un avenir dans le contexte multi devices (tablettes, mobiles...),



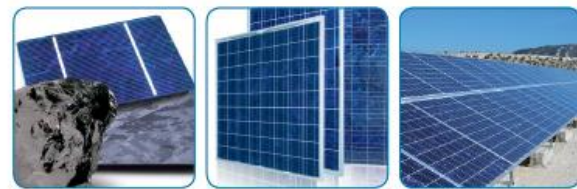
6. Solution SilverLight DotNet

Type Architecture	Architecture répartie - n tiers
Interface utilisateur	Client Web enrichi (RIA)
Langage(s)	SilverLight - C# - LINK
Opérationnel - coefficient 1 <ul style="list-style-type: none"> • Coût par utilisateur : ● • Courbe d'apprentissage : ● • Souplesse de déploiement : ● 	
Performance - coefficient 1 <ul style="list-style-type: none"> • Résistance aux montées en charge potentielles : ● • Délais et rapidité de nouveaux développements : ● • Richesse de l'expérience utilisateur : ● 	
Adaptation au contexte PhotoWatt - coefficient 1 <ul style="list-style-type: none"> • Fonctionnement avec équipe IT réduite : ● • Possibilité de développement en mode silo : ● • Pertinence vis-à-vis de l'infrastructure existante : ● 	
Adaptation au contexte CDSN - coefficient 1 <ul style="list-style-type: none"> • Dialogue avec autres composants (imprimantes...) : ● • Compatibilité avec utilisateur en atelier industriel : ● • Risque de rejets utilisateurs (aversion au changement) : ● 	
Forces <ul style="list-style-type: none"> • Le mode client SilverLight est réputé pour le très beau design et la grande qualité graphique des applications qu'il peut proposer, • Les interfaces Web offrent ergonomie et design premium (très utilisé pour les jeux notamment, 	Faiblesses <ul style="list-style-type: none"> • Le silverlight est assez long à appréhender pour un développeur, car il s'agit avant tout d'un langage à destination des graphistes,
Opportunités <ul style="list-style-type: none"> • Les bénéfices en terme de développement et de code reste les mêmes que pour la solution ASP DotNet, 	Obstacles <ul style="list-style-type: none"> • La technologie Silverlight est en pleine obsolescence, et celle-ci ne sera sans doute pas supportée par les futurs navigateurs Web, • Le mode client Web RIA Silverlight offre de bons rendus mais à l'instar de son homologue Adobe, Flash Player, il nécessite un plug-in IE parfois problématique avec les clients légers (mise à jour, problème de sécurité),

Score total : 10 sur 72 - Pourcentage de scoring : 27,78 %



Conclusion : Comme Adobe Flash Player, cette solution est la preuve parfaite de la rapide obsolescence des technologies de développement Web. Très ancrée, il y a encore 5 ans, sur certains secteurs niches du développement Web (jeux, applications financières et industrielles en temps réel), la technologie est aujourd'hui abandonnée et ne sera plus supportée par les futures versions des principaux navigateurs Web...



7. Solution PHP - ZEND

Type Architecture	Architecture répartie - n tiers
Interface utilisateur	Client Web
Langage(s)	PHP - HTML - CSS
<p>Opérationnel - coefficient 1</p> <ul style="list-style-type: none"> • Coût par utilisateur : ● • Courbe d'apprentissage : ● • Souplesse de déploiement : ● <p>Performance - coefficient 1</p> <ul style="list-style-type: none"> • Résistance aux montées en charge potentielles : ● • Délais et rapidité de nouveaux développements : ● • Richesse de l'expérience utilisateur : ● <p>Forces</p> <ul style="list-style-type: none"> • Le framework Zend permet des gains en temps de développement en raison de la mise à disposition de toutes les briques génériques (authentification, navigation), <p>Opportunités</p> <ul style="list-style-type: none"> • La technologie PHP est assez ancienne et il existe de très nombreux développeurs, d'où un taux journalier plutôt abordable, 	
<p>Adaptation au contexte PhotoWatt - coefficient 1</p> <ul style="list-style-type: none"> • Fonctionnement avec équipe IT réduite : ● • Possibilité de développement en mode silo : ● • Pertinence vis-à-vis de l'infrastructure existante : ● <p>Adaptation au contexte CDSN - coefficient 1</p> <ul style="list-style-type: none"> • Dialogue avec autres composants (imprimantes...) : ● • Compatibilité avec utilisateur en atelier industriel : ● • Risque de rejets utilisateurs (aversion au changement) : ● <p>Faiblesses</p> <ul style="list-style-type: none"> • Le PHP est destiné aux sites Web mais s'avère peu pertinent pour les applications industrielles, <p>Obstacles</p> <ul style="list-style-type: none"> • Le dialogue avec les API système (Win32) et les drivers d'imprimante peut s'avérer complexe avec PHP car ce n'est pas son usage premier, 	

Score total : 11 sur 72 - Pourcentage de scoring : 30,56 %



Conclusion : Un projet PHP avec Framework Zend est parfait pour le déploiement d'un site Web mais n'a pas de réelle pertinence pour un projet d'application de suivi de traçabilité en environnement industriel.



8. Solution WinDev PC soft

Type Architecture	Architecture hybride
Interface utilisateur	Navigateur Web - Client lourd - Application native
Langage(s)	WLangage - SQL
<p>Opérationnel - coefficient 1</p> <ul style="list-style-type: none"> • Coût par utilisateur : ● • Courbe d'apprentissage : ● • Souplesse de déploiement : ● <p>Performance - coefficient 1</p> <ul style="list-style-type: none"> • Résistance aux montées en charge potentielles : ● • Délais et rapidité de nouveaux développements : ● • Richesse de l'expérience utilisateur : ● 	
<p>Adaptation au contexte PhotoWatt - coefficient 1</p> <ul style="list-style-type: none"> • Fonctionnement avec équipe IT réduite : ● • Possibilité de développement en mode silo : ● • Pertinence vis-à-vis de l'infrastructure existante : ● <p>Adaptation au contexte CDSN - coefficient 1</p> <ul style="list-style-type: none"> • Dialogue avec autres composants (imprimantes...) : ● • Compatibilité avec utilisateur en atelier industriel : ● • Risque de rejets utilisateurs (aversion au changement) : ● 	
<p>Forces</p> <ul style="list-style-type: none"> • La partie Front-End est très rapide et facile à programmer et permet de réaliser des projets multi plateformes en très peu de temps. De même, les modifications et ajustements sont aisés, • Les licences sont plutôt bons marchés et s'avère moins couteuse que Visual Studio (les PME sont le coeur de cible), <p>Le langage (WLangage) de base utilisé est un dérivé du Visual Basic qui ne nécessite que quelques jours d'apprentissage pour un développeur non spécialisé,</p> <ul style="list-style-type: none"> • Très souple, elle combine les avantages du mode Full Web (une programmation d'écran pour plusieurs périphériques différents), et les bénéfices du client lourd (performances accrues et dialogue avec les composants systèmes), 	<p>Faiblesses</p> <ul style="list-style-type: none"> • Le produit est français et n'est pas utilisé aux Etats Unis, ce qui peut être un frein pour une multi nationale (ce qui n'est pas encore le cas pour PhotoWatt),
<p>Opportunités</p> <ul style="list-style-type: none"> • La plateforme dispose de nombreux connecteurs performants et prêts à l'emploi vers les différents ERP du marché (dont notamment ceux vers les plateformes SAP), • La plateforme s'intègre parfaitement bien au sein d'une infrastructure Full Microsoft (intégration très facile avec SQL Server et les outils Office, dialogue avec les drivers d'imprimantes ou divers automates), • La plateforme Windev permet aux développeurs de se concentrer sur la partie fonctionnelle et les différentes règles de gestion de leurs applications car tous les composants techniques génériques sont déjà intégrés. Ce produit est donc très adapté pour les PME, car cela leur évite de réinventer la roue, 	<p>Obstacles</p> <ul style="list-style-type: none"> • Le produit n'est pas forcément adapté pour les très grosses applications dépassant les milliers d'utilisateurs (ce qui n'est pas encore le cas de PhotoWatt),

Score total : 32 sur 36 - Pourcentage de scoring : 88,89 %

Conclusion : La plateforme Windev est très adaptée pour les PME industrielles comme PhotoWatt, qui ont des besoins métier spécifiques et des budgets limités... Cette plateforme combine les avantages d'Access (facilité, souplesse et rapidité de développement) avec les plus values des applications Web (gestion des multi-périphériques en Front End, souplesse de déploiement, architecture découplée en services, compatibilité avec les composants système récents, dialogue avec tout type de drivers), sans avoir ses défauts (obsolescence technique, lenteurs, raréfaction des prestataires...),



9. Solution Xamarin Forms

Type Architecture	Architecture hybride
Interface utilisateur	Navigateur Web - Client lourd - Application native
Langage(s)	C# - XAML - SQL

Opérationnel - coefficient 1

- Coût par utilisateur : ●
- Courbe d'apprentissage : ●
- Souplesse de déploiement : ●

Performance - coefficient 1

- Résistance aux montées en charge potentielles : ●
- Délais et rapidité de nouveaux développements : ●
- Richesse de l'expérience utilisateur : ●

Adaptation au contexte PhotoWatt - coefficient 1

- Fonctionnement avec équipe IT réduite : ●
- Possibilité de développement en mode silo : ●
- Pertinence vis-à-vis de l'infrastructure existante : ●

Adaptation au contexte CDSN - coefficient 1

- Dialogue avec autres composants (imprimantes...) : ●
- Compatibilité avec utilisateur en atelier industriel : ●
- Risque de rejets utilisateurs (aversion au changement) : ●

Forces

- La partie Front-End est très rapide et facile à programmer et permet de réaliser des projets multi plateformes en très peu de temps. De même, les modifications et ajustements sont aisés,

Opportunités

- La plateforme s'intègre parfaitement bien au sein d'une infrastructure Full Microsoft puisque Xamarin a été racheté par l'éditeur en 2016,

Faiblesses

- Les licences Xamarin sont très onéreuses et s'amortissent généralement sur un nombre d'utilisateurs important (plusieurs milliers) dans le contexte d'applications déployées dans des stores à l'échelle mondiale,

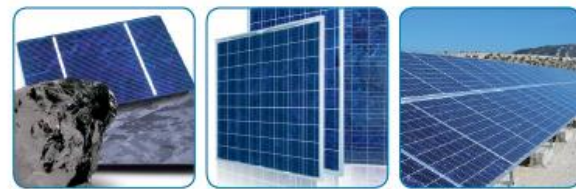
Obstacles

- Le produit est plutôt orienté application mobile et est sans doute over designé pour une application comme CDSN,

Score total : 25 sur 36 - Pourcentage de scoring : 69,44 %



Conclusion : La plateforme est techniquement séduisante mais peut être plus adaptée à un éditeur logiciel pure player dont la priorité serait les périphériques mobiles (et non un complément comme dans le cas de PhotoWatt). De plus les licences auront du mal à rentrer dans les budgets car l'amortissement de ce produit se fait généralement sur l'échelle de milliers d'utilisateurs,



10. Solution WPF - DotNet

Type Architecture	Architecture client mono poste
Interface utilisateur	Interface Windows Form
Langage(s)	Visual Basic - XAML - LINQ

Opérationnel - coefficient 1

- Coût par utilisateur : ●
- Courbe d'apprentissage : ●
- Souplesse de déploiement : ●

Performance - coefficient 1

- Résistance aux montées en charge potentielles : ●
- Délais et rapidité de nouveaux développements : ●
- Richesse de l'expérience utilisateur : ●

Adaptation au contexte PhotoWatt - coefficient 1

- Fonctionnement avec équipe IT réduite : ●
- Possibilité de développement en mode silo : ●
- Pertinence vis-à-vis de l'infrastructure existante : ●

Adaptation au contexte CDSN - coefficient 1

- Dialogue avec autres composants (imprimantes...) : ●
- Compatibilité avec utilisateur en atelier industriel : ●
- Risque de rejets utilisateurs (aversion au changement) : ●

Forces

- La solution permet de bénéficier des fonctionnalités avancées des langages C# et Visual Basic (design pattern, héritage), ce qui permet de créer des composants génériques réutilisables,
- La possibilité de structurer fonctionnellement le code (modélisation objet) facilite les travaux d'évolution applicative,

Opportunités

- La communication avec les composants système (imprimantes, drivers, automates) et les API Windows (Win32) est facile et complètement native,
- La programmation WPF (Windows Platform Foundation) par formulaire utilise le XAML, un mode de développement graphique en drag & drop, qui est rapide et est aussi facile à utiliser qu'Access,

Faiblesses

- La solution en client lourd ne permet pas un déploiement sur des périphériques mobiles,

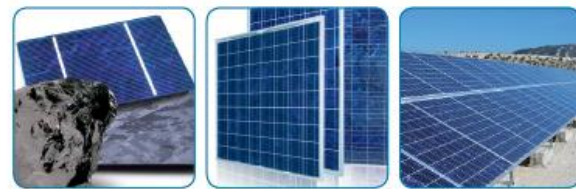
Obstacles

- Les interfaces utilisateur restent d'un design et d'une ergonomie plus limitée qu'en mode Web,
- L'architecture client lourd pose les mêmes problématiques de déploiement et de maintenance Hotline que pour Access (déploiement sur les postes utilisateur),

Score total : 26 sur 72 - Pourcentage de scoring : 72,22 %



Conclusion : Le projet en mode client lourd XAML offre une valeur ajoutée potentielle par rapport à un projet Access, tout en conservant ses avantages. De plus, la courbe d'apprentissage en interne serait faible et les développements d'écran seraient faciles et relativement rapides ce qui permettrait d'envisager un projet CDSN à moindre coût et à délais resserrés,



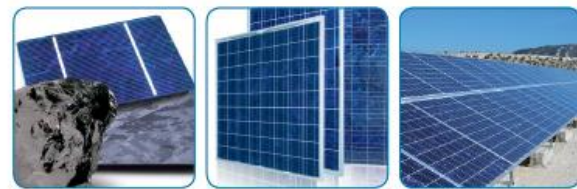
11. Solution Access mode ADP

Type Architecture	Architecture client lourd - mono poste
Interface utilisateur	Access
Langage(s)	Transact SQL - VBA
<p>Opérationnel - coefficient 1</p> <ul style="list-style-type: none"> • Coût par utilisateur : ● • Courbe d'apprentissage : ● • Souplesse de déploiement : ● <p>Performance - coefficient 1</p> <ul style="list-style-type: none"> • Résistance aux montées en charge potentielles : ● • Délais et rapidité de nouveaux développements : ● • Richesse de l'expérience utilisateur : ● 	
<p>Adaptation au contexte PhotoWatt - coefficient 1</p> <ul style="list-style-type: none"> • Fonctionnement avec équipe IT réduite : ● • Possibilité de développement en mode silo : ● • Pertinence vis-à-vis de l'infrastructure existante : ● <p>Adaptation au contexte CDSN - coefficient 1</p> <ul style="list-style-type: none"> • Dialogue avec autres composants (imprimantes...) : ● • Compatibilité avec utilisateur en atelier industriel : ● • Risque de rejets utilisateurs (aversion au changement) : ● 	
<p>Forces</p> <ul style="list-style-type: none"> • Access en mode ADP permet de découpler toute la gestion des données sur SQL Server (plus de limitation à 255 champs), • Les objets SQL Server (procédures stockées) sont plus facilement réutilisables et seul le langage Transact SQL conforme à la norme ANSI est utilisé (contrairement au Jet SQL), • Les temps de traitement sont améliorés car les données ne sont pas retraitées par le moteur Jet d'Access , • Les performances du middleware OLE DB sont meilleures que celles du connecteur ODBC (gestion erreurs connexion facilitée), 	<p>Faiblesses</p> <ul style="list-style-type: none"> • L'application reste en mode client lourd, avec toutes les contraintes de déploiement et de maintenance associées, • Il n'est pas possible d'automatiser les tests unitaires, • Les limitations du langage VBA restent les mêmes que pour un projet Access classique,
<p>Opportunités</p> <ul style="list-style-type: none"> • ADP permet de s'affranchir des connecteurs ODBC capricieux, • Les connexions en mode Authentification Windows permettent d'éviter les connexions utilisateur anonymes, • Le meilleur découplage de l'application permet d'envisager une migration ultérieure en Dot Net, 	<p>Obstacles</p> <ul style="list-style-type: none"> • Le mode Access ADP n'est plus supportée à partir de la version 2013 d'Access,

Score total : 17 sur 72 - Pourcentage de scoring : 47,22 %



Conclusion : Le projet Access en mode ADP peut être vu comme un compromis entre Access et DotNet et peut constituer une étape intermédiaire avant une migration ultérieure vers une architecture en mode client Web. Mais au vu de son incompatibilité avec les nouvelles versions d'Access, il n'est pas raisonnable d'envisager techniquement cette solution...



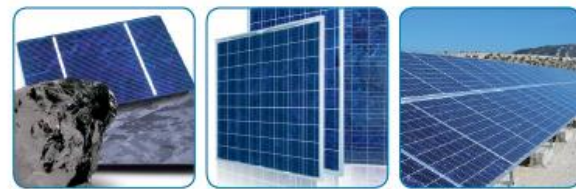
12. Solution Microsoft Access

Type Architecture	Architecture client mono poste
Interface utilisateur	Access
Langage(s)	Jet SQL - VBA
<p>Opérationnel - coefficient 1</p> <ul style="list-style-type: none"> • Coût par utilisateur : ● • Courbe d'apprentissage : ● • Souplesse de déploiement : ● <p>Performance - coefficient 1</p> <ul style="list-style-type: none"> • Résistance aux montées en charge potentielles : ● • Délais et rapidité de nouveaux développements : ● • Richesse de l'expérience utilisateur : ● 	
<p>Adaptation au contexte PhotoWatt - coefficient 1</p> <ul style="list-style-type: none"> • Fonctionnement avec équipe IT réduite : ● • Possibilité de développement en mode silo : ● • Pertinence vis-à-vis de l'infrastructure existante : ● <p>Adaptation au contexte CDSN - coefficient 1</p> <ul style="list-style-type: none"> • Dialogue avec autres composants (imprimantes...) : ● • Compatibilité avec utilisateur en atelier industriel : ● • Risque de rejets utilisateurs (aversion au changement) : ● 	
<p>Forces</p> <ul style="list-style-type: none"> • Les délais de création d'un projet sous Access peuvent être relativement courts et peu onéreux, 	<p>Faiblesses</p> <ul style="list-style-type: none"> • Il n'est pas possible d'automatiser les tests unitaires, • Les connecteurs ODBC présentent de nombreuses contraintes (lourdeur d'installation, anonymat des connexions), • Le moteur de BDD Jet d'Access est peu performant, • Toute requête SQL Server doit être réécrite en Jet SQL • Les capacités de VBA restent limitées (peu d'API disponibles, modélisation objet embryonnaire, pas de gestion d'exceptions),
<p>Opportunités</p> <ul style="list-style-type: none"> • Le process de mise en production d'un projet Access est bien maîtrisé en interne, 	<p>Obstacles</p> <ul style="list-style-type: none"> • Le VBA est limité et ne permet pas de profiter des outils de programmation moderne et des gains de productivité afférents (modélisation objet, framework, réutilisation de composants), • Le déploiement et la maintenance d'application en mode client lourd est consommateur de ressources Hotline,

Score total : 16 sur 72 - Pourcentage de scoring : 44,44 %



Conclusion : Le projet Access permet d'obtenir des temps de développement intéressants à court terme, mais ces gains s'altèrent à moyen terme à mesure que l'application doit être modifiée, là où justement les possibilités de factorisation de code des langages plus évolués permettent des gains de productivité conséquents (découplage des couches applicatives, API prêtes à l'emploi, modélisation objet proche du modèle fonctionnel, réutilisation de composants génériques),



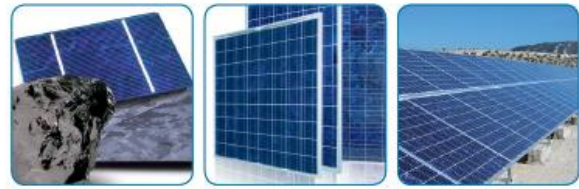
13. Solution MES ERP

Type Architecture	Architecture ERP propriétaire
Interface utilisateur	Ecrans SAP
Langage(s)	ABAP
<p>Opérationnel - coefficient 1</p> <ul style="list-style-type: none"> • Coût par utilisateur : ● • Courbe d'apprentissage : ● • Souplesse de déploiement : ● <p>Performance - coefficient 1</p> <ul style="list-style-type: none"> • Résistance aux montées en charge potentielles : ● • Délais et rapidité de nouveaux développements : ● • Richesse de l'expérience utilisateur : ● 	
<p>Adaptation au contexte PhotoWatt - coefficient 1</p> <ul style="list-style-type: none"> • Fonctionnement avec équipe IT réduite : ● • Possibilité de développement en mode silo : ● • Pertinence vis-à-vis de l'infrastructure existante : ● <p>Adaptation au contexte CDSN - coefficient 1</p> <ul style="list-style-type: none"> • Dialogue avec autres composants (imprimantes...) : ● • Compatibilité avec utilisateur en atelier industriel : ● • Risque de rejets utilisateurs (aversion au changement) : ● 	
<p>Forces</p> <ul style="list-style-type: none"> • L'ERP permet de bénéficier de briques standards prêtes à l'emploi, qui ont été enrichies et améliorées au fil des années par de multiples remontées utilisateur évoluant dans des contextes industriels concurrentiels, • Cette solution permet de ne pas avoir à réinventer la roue sur des besoins génériques et standards, 	<p>Faiblesses</p> <ul style="list-style-type: none"> • Le coût des licences et des consultants SAP peuvent très vite faire gonfler la facture du projet, • Le reporting serait sans à refaire (passage à Business Object), ce qui augmentera également les coûts éventuels,
<p>Opportunités</p> <ul style="list-style-type: none"> • L'une des grosses plus values serait la complète harmonisation des données ERP et traçabilité, ce qui simplifierait les reporting de l'UAP Wafers (actuellement les données CDSN et Baan ne se recoupent pas toujours), • Le cadre structurant imposé par l'ERP pourrait être bénéfique sur les méthodes de travail (le développement spécifique permettant justement le contraire), • Le MES permettrait de mieux encadrer les dépenses de développement sur la traçabilité (l'horizon des possibles n'inspire pas la frugalité des demandes utilisateur), 	<p>Obstacles</p> <ul style="list-style-type: none"> • La solution du MES n'est envisageable que si un nouvel ERP étant mis en place dans la nouvelle entité Crystal Advanced, • La personnalisation de l'outil de traçabilité est beaucoup moins flexible et généralement il est nécessaire de rester sur le périmètre des besoins utilisateurs standards,

Score total : 28 sur 72 - Pourcentage de scoring : 77,78 %



Conclusion : Dans l'hypothèse d'un déploiement SAP, la solution du MES peut s'avérer pertinente dans le cadre de PhotoWatt mais nécessiterait une étude plus approfondie pour évaluer les coûts potentiels et la couverture fonctionnelle des besoins utilisateur,



-

5. RESULTATS DE L'ETUDE

5.1 Synthèse des scoring des solutions

Le calcul des différents scoring s'est fait selon des critères de :

- Performances techniques,
- Pertinence fonctionnelle,
- Adaptation au contexte PhotoWatt,
- Problématique de coût et de délai de développement,

Après calcul de ces différents scoring, il s'avère que les solutions qui présentent les meilleurs résultats à l'aune de ces différents critères sont :

- La solution WinDev de PC Soft
 - Des coûts de développement faible et des délais rapide,
 - Un déploiement possible sur téléphones et tablettes sans développement complémentaire excessif,
 - Une bonne communication avec les drivers d'imprimantes,
 - Une très bonne intégration avec les produits Office (Word, Excel),
 - De bonnes performances avec les bases de données SQL Server,
 - Des connecteurs avec les ERP SAP de très bonne qualité,
- La solution MES dans le cadre d'une intégration d'un nouvel ERP,
 - Synergie possible avec l'ERP,
 - Des coûts de paramétrage moins important que des coûts de développement,
 - Des plus-values sur le reporting,
- La solution Microsoft WPF (Window Platform Foundation)
 - Des développements faciles et rapides,
 - L'intégration au sein de la plateforme DotNet,



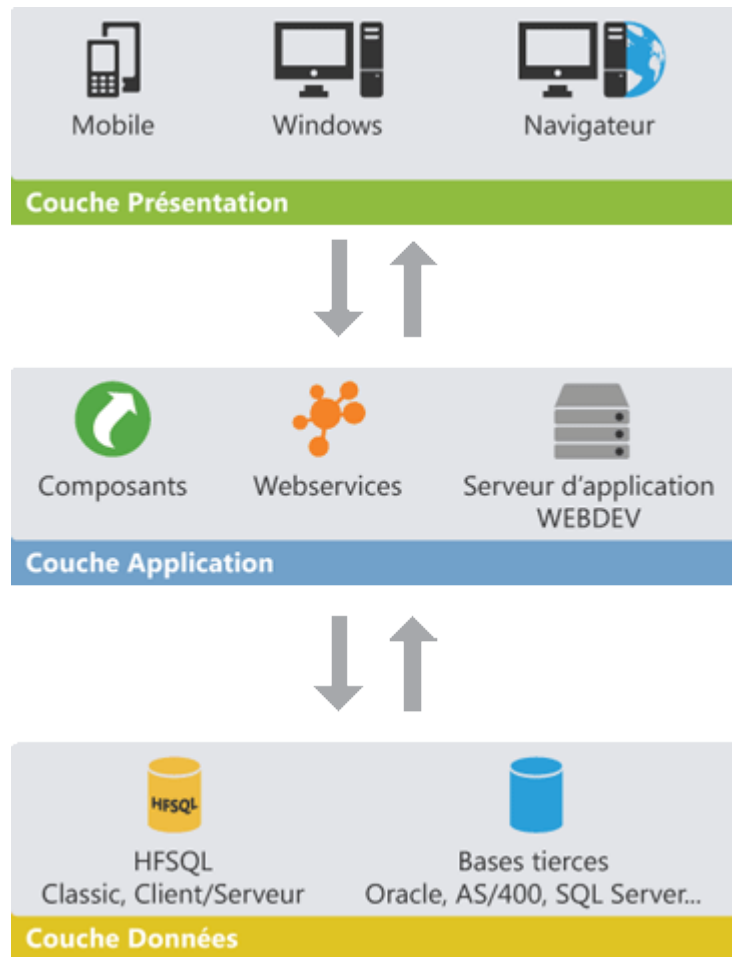
Présentation des résultats :

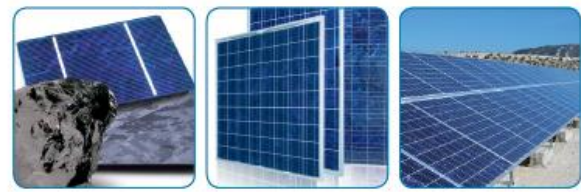
Solutions potentielles	Scoring total	Score technique	Score contextuel	Commentaires
8. Solution WinDev	1 88,89%	88,89%	88,89%	Coût faible - développement facile et rapide -déploiement Web, tablette & clients lourds (imprimantes)
13. Solution MES ERP	2 77,78%	88,89%	66,67%	Synergie avec l'ERP - Facilité de la maintenance
10. Solution WPF -DotNet	3 72,22%	55,56%	88,89%	Développement facile et rapide - mais uniquement clients lourds (pas de tablette ou de Web)
9. Solution Xamarin	4 69,44%	72,22%	66,67%	Solution onéreuse et plutôt destinée aux applications purement mobiles
5. Solution ASP DotNet	5 52,78%	66,67%	38,89%	Développements pages Web assez longs - ni tablette ni mobile - communications imprimantes moins aisées
11. Solution Access ADP	6 47,22%	50,00%	44,44%	Solution abandonnée par Microsoft malgré d'intéressantes plus values dans le contexte PWT
3. Solution App devices mobiles	7 44,44%	33,33%	55,56%	Solution trop exigeante en terme de budget et de ressources
12. Solution Access	8 44,44%	38,89%	50,00%	Solution de secours si jamais aucun budget n'était disponible
4. Solution J2EE	9 41,67%	44,44%	38,89%	Environnement Java non pertinent pour l'infrastructure PWT
1. Solution Full Web	10 38,89%	72,22%	5,56%	Très bonne solution technique mais non adaptée au contexte de l'application CDSN
2. Solution Full Web Microsoft	11 38,89%	55,56%	22,22%	Solution trop exigeante techniquement au vu des délais et des ressources internes
7. Solution PHP	12 30,56%	44,44%	16,67%	A n'envisager que pour des sites Web
6. Solution SilverLight DotNet	13 27,78%	44,44%	11,11%	Solution devenue techniquement obsolète malgré d'indéniables avantages



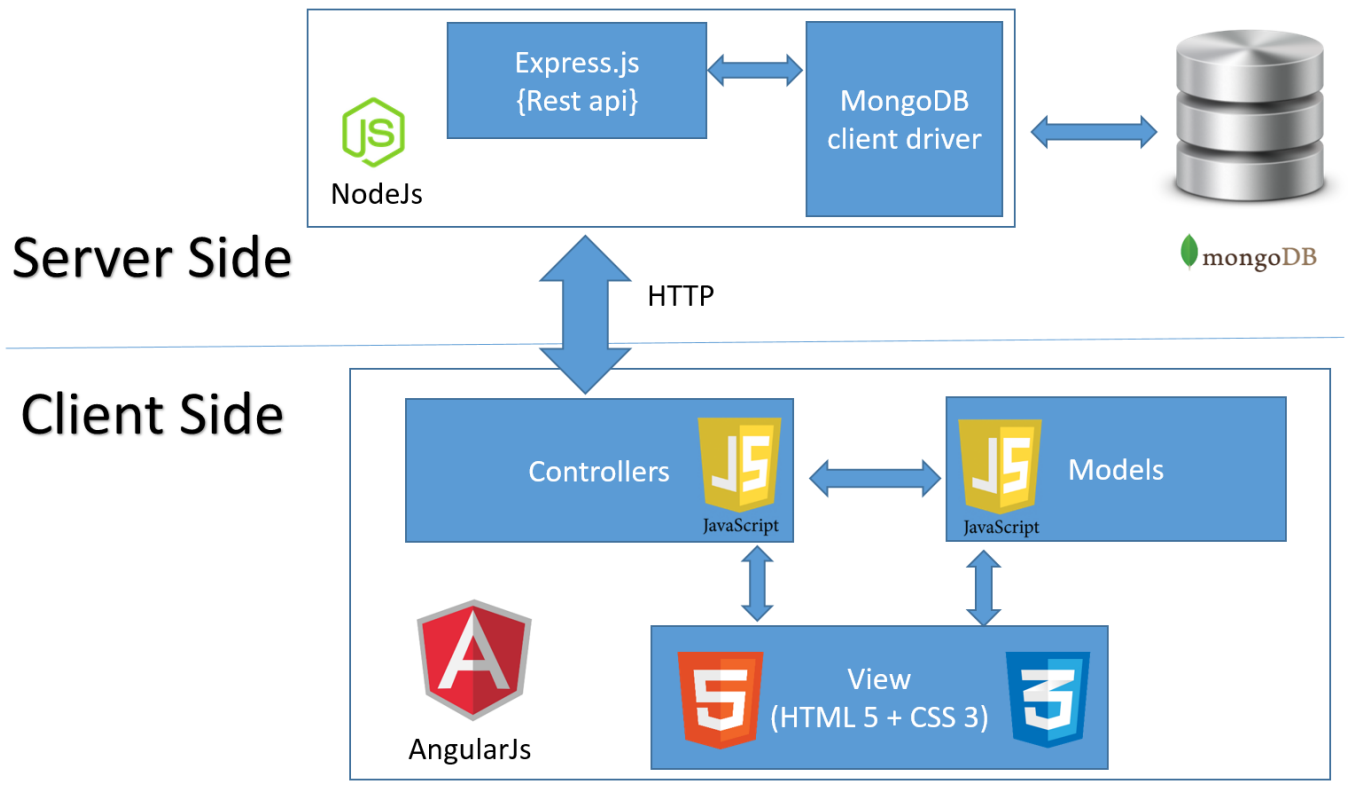
5.2 Annexes

Architecture WinDev :





Architecture Full Web avec Angular JS :





Architecture en mode Mobile devices :

