



Document d'architecture d'entreprise

Transformation vers une architecture distribuée

LOUIS ZEPHIR

Historique

<i>Auteur</i>	<i>Remarques</i>	<i>Date</i>	<i>n° version</i>
<i>Louis ZEPHIR</i>	Création du documents, début de rédaction	01/02/2026	1.0

Auteurs

<i>Auteur</i>	<i>Fonction</i>	<i>Contact</i>
<i>Louis ZEPHIR</i>	Architecte Logiciel	louis.zephir@websonic.fr

Table des matières

PHASE A : VISION D'ARCHITECTURE	3
Contexte métier et éléments moteurs	3
Problématiques Métier	3
Objectifs Stratégiques	3
Périmètre du travail d'architecture.....	3
Parties prenantes et Préoccupations.....	3
PHASE B : ARCHITECTURE MÉTIER	5
Architecture métier actuelle (AS-IS)	5
Processus actuel.....	5
Limitations identifiées	5
Architecture métier cible (TO-BE)	6
Processus cible	6
Capacités Métier Cibles.....	6
PHASE C : ARCHITECTURE DES SI	7
Architecture des données	7
Principe de séparation des données	7
Architecture des applications	7
Building Blocks Applicatifs (ABBs)	7
Interactions applicatives.....	7
PHASE D : ARCHITECTURE TECHNOLOGIQUE.....	7
Composants technologiques	7
Infrastructure WebSonic	7
Infrastructure client	8
Principes d'architecture	8
Sécurité et Gouvernance	8
Contrôles de Sécurité	8
Matrice des droits d'accès	8
PHASE E : ANALYSE GAP ET SOLUTIONS.....	9
Analyse des écarts (Gap Analysis)	9
Building Blocks de Solution (SBBs)	9
Composants à développer	9
Composants existants à adapter	9
Stratégie de mise en œuvre	9
Approche de transition	9
Phases d'implémentation	10
Contraintes et Risques	10
CONCLUSION ET RECOMMANDATIONS	11
Bénéfices Attendus	11
Prochaines Étapes.....	11

PHASE A : VISION D'ARCHITECTURE

Contexte métier et éléments moteurs

WebSonic est une agence de création de sites web utilisant une approche par templates pour livrer rapidement des sites personnalisés à ses clients internationaux.

La croissance récente des ventes mondiales a mis en évidence les limites de l'architecture actuelle.

Problématiques Métier

- **Surcharge des équipes** : Les équipes de développement sont submergées par le volume de demandes
- **Complexité logicielle** : L'environnement logiciel n'a pas évolué avec l'entreprise
- **Souveraineté des données** : Les clients exigent que leurs données restent sur leurs propres serveurs
- **Délais de livraison** : 72 heures actuellement pour un cycle complet de développement

Objectifs Stratégiques

1. Réduire le temps de cycle de développement de 72h à 36h (50%)
2. Réduire le nombre de transferts inter-équipes de 6 à 3 (50%)
3. Garantir la souveraineté des données clients
4. Améliorer l'agilité et la communication des équipes de développement
5. Créer une plateforme évolutive pour l'innovation continue

Périmètre du travail d'architecture

Largeur : Processus de développement et déploiement de sites web clients, incluant les équipes de développement WebSonic et l'infrastructure technique.

Profondeur : Architecture applicative (templates et modules), architecture de données (séparation WebSonic/Clients), architecture technologique (serveurs, agents, API).

Période : Architecture cible à implémenter en 6 mois, avec vision à 2 ans pour évolution.

Domaines d'architecture : Business, Application, Data, Technology

Parties prenantes et Préoccupations

Jeanne (CEO)	Croissance entreprise, compétitivité, ROI	Haute - Sponsor exécutif
Cathy (CTO)	Faisabilité technique, évolutivité, sécurité	Haute - Sponsor technique

Louis (COO)	Continuité opérationnelle, respect échéances	Haute - Garant opérations
Louis Z (AL)	Cohérence architecturale, faisabilité technique, conformité TOGAF, qualité des livrables	Haute - Responsable de l'architecture (Architecte référent")
Agnès & Tim (Managers)	Efficacité équipes, réduction transferts, outils	Moyenne - Utilisateurs clés
Équipes Dév.	Simplicité d'usage, documentation, standardisation	Moyenne - Utilisateurs finaux
Clients	Souveraineté données, disponibilité, personnalisation	Haute - Bénéficiaires finaux

PHASE B : ARCHITECTURE MÉTIER

Architecture métier actuelle (AS-IS)

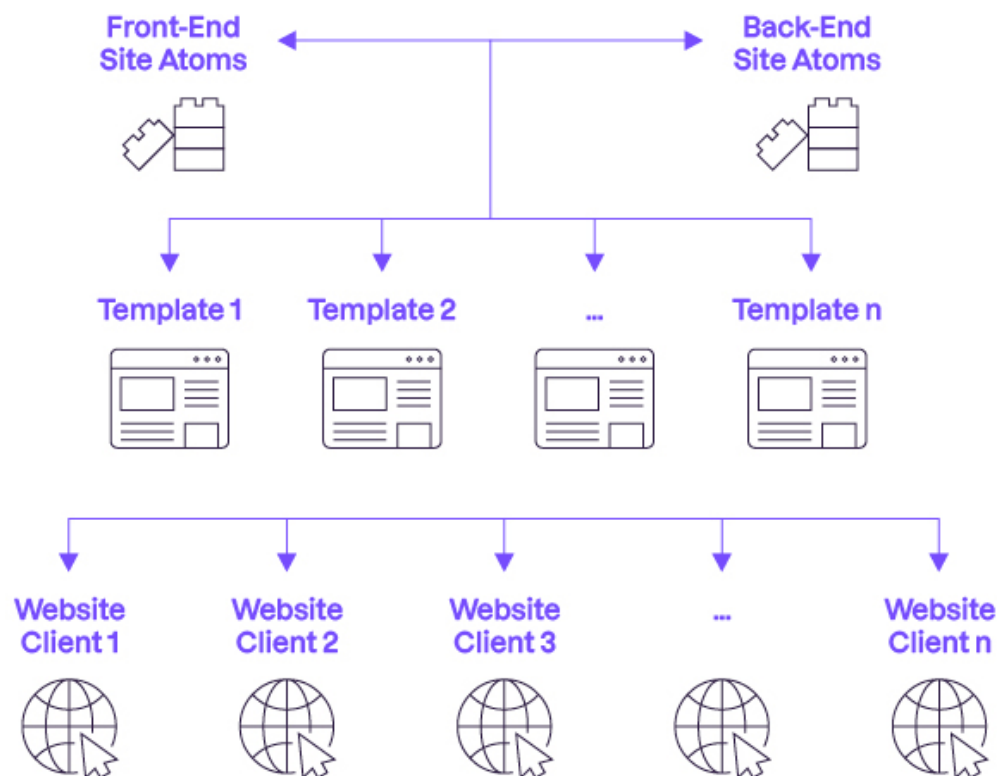
Modèle opérationnel : Modèle centralisé avec développement séquentiel par équipe principale, puis transferts aux équipes secondaires selon les fuseaux horaires.

Processus actuel

1. Client commande un site web
2. Équipe principale débute le développement avec ses propres templates
3. Transfert à équipe secondaire en fin de journée (si incomplet)
4. Cycle se répète jusqu'à 6 transferts
5. Validation complète requise par équipe principale
6. Déploiement sur serveurs WebSonic (72h en moyenne)

Limitations identifiées

- **Fragmentation** : Chaque développeur utilise ses propres templates
- **Retards** : Les transferts de fichiers entre équipes causent des délais
- **Centralisation** : Toutes les données clients hébergés chez WebSonic
- **Scalabilité** : Architecture non conçue pour la croissance actuelle



[Voir le schéma](#)

Architecture métier cible (TO-BE)

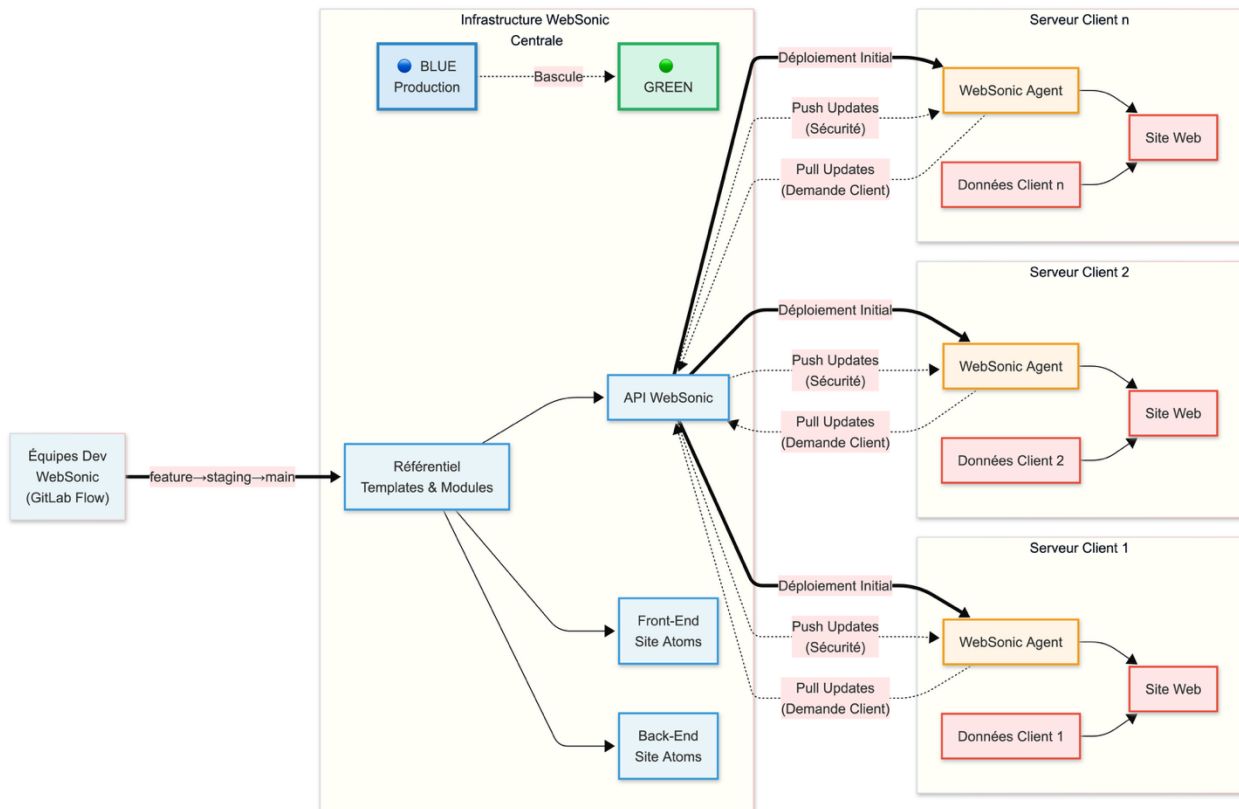
Modèle opérationnel cible : Modèle distribué avec référentiel centralisé de templates et déploiement automatisé sur serveurs clients.

Processus cible

1. Client commande un site web
2. Sélection templates et modules depuis référentiel centralisé
3. Personnalisation via paramètres (langue, couleur, style)
4. WebSonic Agent déploie automatiquement sur serveur client
5. Validation finale et mise en production (36h cible)
6. Mises à jour gérées via Agent (Push/Pull)

Capacités Métier Cibles

- **Gestion centralisée des templates :** Repository unique avec versioning
- **Déploiement automatisé :** Via WebSonic Agent sur serveurs clients
- **Personnalisation paramétrique :** Sans code custom, via configuration
- **Mises à jour bidirectionnelles :** Push (sécurité) et Pull (demandes)
- **Monitoring et gouvernance :** Suivi temps réel des déploiements



[Voir le schéma](#)

PHASE C : ARCHITECTURE DES SI

Architecture des données

Principe de séparation des données

L'architecture de données respecte une séparation stricte selon le principe de souveraineté :

Catégorie	Localisation	Exemples
Données Structurelles	Serveurs WebSonic	Templates, Modules, Code front/back-end
Données Clients	Serveurs Clients	Contenu, Utilisateurs, Config, Assets
Données Métadonnées	Serveurs WebSonic	Versions, Logs déploiement, Métriques

Architecture des applications

Building Blocks Applicatifs (ABBs)

1. **Référentiel de Templates** : Stockage centralisé et versionné des templates Front-End et Back-End
2. **Référentiel de Modules (Site Atoms)** : Bibliothèque de composants réutilisables (login, header, formulaires, etc.)
3. **API WebSonic** : Interface REST pour orchestration des déploiements et mises à jour
4. **WebSonic Agent** : Composant logiciel installé côté client pour synchronisation et déploiement
5. **Système de Versioning** : Gestion des versions avec tags et rollback
6. **Monitoring & Gouvernance** : Tableaux de bord et audit logs

Interactions applicatives

- Équipes de développement ↔ Référentiel Templates/Modules (CRUD)
- API WebSonic ↔ WebSonic Agents (Push updates, Pull requests)
- WebSonic Agent ↔ Serveur Client (Déploiement local)
- Système Versioning ↔ Référentiel (Gestion état)

PHASE D : ARCHITECTURE TECHNOLOGIQUE

Composants technologiques

Infrastructure WebSonic

- **Serveurs d'application** : Cluster haute disponibilité pour héberger le référentiel et l'API

- **Base de données** : PostgreSQL pour métadonnées, Git pour versioning des templates
- **Stockage** : Object storage (S3-compatible) pour templates binaires et assets
- **Load Balancer** : Distribution des requêtes API avec SSL/TLS termination

Infrastructure client

- **WebSonic Agent** : Application containerisée (Docker) ou service système
- **Web Server** : Nginx ou Apache pour servir le site web
- **Base de données** : MySQL/PostgreSQL pour données clients (au choix du client)
- **Certificats SSL** : Let's Encrypt ou certificats fournis par client

Principes d'architecture

1. **Séparation des préoccupations** : WebSonic gère le code, clients gèrent les données
2. **Réutilisabilité** : Templates et modules standardisés pour tous les clients
3. **Sécurité par défaut** : Authentification mutuelle TLS, principe du moindre privilège
4. **Automatisation** : Déploiement et mises à jour sans intervention manuelle
5. **Évolutivité horizontale** : Ajout de serveurs selon croissance sans reengineering
6. **Traçabilité** : Audit complet de toutes les opérations
7. **Résilience** : Capacité de rollback et récupération en cas d'échec

Sécurité et Gouvernance

Contrôles de Sécurité

- **Authentification** : Certificats client TLS pour agents, OAuth2/OIDC pour utilisateurs internes
- **Autorisation** : RBAC (Role-Based Access Control) avec permissions granulaires
- **Chiffrement** : TLS 1.3 pour transit, chiffrement at-rest pour données sensibles
- **Intégrité** : Signatures numériques SHA-256 pour templates et modules
- **Audit** : Logs centralisés immuables avec rétention 7 ans

Matrice des droits d'accès

Rôle	Templates	API/Agents	Données client
CTO	R/W/D	Admin	Métriques (R)
Managers	R/W (scope)	Deploy Test	Aucun
Développeurs	R/W (assigné)	Logs (R)	Aucun
Clients	Aucun	Pull Update	R/W/D (Total)

PHASE E : ANALYSE GAP ET SOLUTIONS

Analyse des écarts (Gap Analysis)

Domaine	Écart identifié	Solution proposée
Templates	Fragmentation, absence de référentiel centralisé	Créer référentiel Git centralisé avec CI/CD
Déploiement	Processus manuel, lent, avec transferts multiples	Développer WebSonic Agent pour automatisation
Données	Centralisation chez WebSonic, non conforme souveraineté	Hébergement données sur serveurs clients uniquement
Mises à jour	Aucun mécanisme standardisé de mise à jour	Implémenter système Push/Pull via API
Gouvernance	Absence de traçabilité et audit	Mettre en place logging centralisé et RBAC

Building Blocks de Solution (SBBs)

Composants à développer

1. **WebSonic Agent (Priorité Haute)** : Application client pour synchronisation et déploiement automatisé
2. **API REST WebSonic (Priorité Haute)** : Endpoints pour authentification, versioning, déploiement, monitoring
3. **Référentiel Git (Priorité Haute)** : GitLab/GitHub Enterprise pour versioning templates et modules
4. **Système de Signature (Priorité Haute)** : PKI pour signature numérique des templates
5. **Dashboard Monitoring (Priorité Moyenne)** : Interface web pour KPIs et statuts déploiements
6. **Système de Rollback (Priorité Moyenne)** : Mécanisme de retour arrière automatique en cas d'échec

Composants existants à adapter

- Templates actuels : Migration vers référentiel Git avec standardisation
- Processus de développement : Adaptation pour utiliser référentiel centralisé
- Documentation : Refonte complète pour approche globale (équipes internationales)

Stratégie de mise en œuvre

Approche de transition

Migration progressive "Big Bang" évitée : L'architecture actuelle reste opérationnelle pour clients existants pendant que la nouvelle architecture est déployée pour nouveaux clients uniquement.

Phases d'implémentation

Phase	Durée	Livrables clés	Critères de succès
Phase 1	Mois 1-2	Agent WebSonic, API REST, PKI	Tests unitaires 90%+ coverage
Phase 2	Mois 3-4	Migration référentiel, 3 clients pilotes	Déploiement < 36h validé
Phase 3	Mois 5-6	Déploiement nouveaux clients, monitoring	10+ clients en production
Phase 4	Mois 7+	Migration clients existants (opt.), RTL support	Réduction transferts à 3 validée

1 [Voir la feuille de la route](#) – 2 [Voir le Trello](#) – 3 [Voir matrice RACI](#) – 4 [Voir budget du projet](#)

Contraintes et Risques

- **Contrainte** : Continuité de service requise → Migration progressive sans interruption
- **Risque** : Complexité agent client → Mitigation : expertise externe, tests approfondis
- **Risque** : Adoption équipes → Mitigation : formation, documentation, support dédié
- **Risque** : Sécurité agents → Mitigation : authentification TLS mutuelle, audit externe

[Voir tableau de prévention des risques](#)

CONCLUSION ET RECOMMANDATIONS

Cette architecture d'entreprise conforme TOGAF répond aux objectifs stratégiques de WebSonic en proposant une transformation mesurée et réaliste de l'architecture actuelle vers un modèle distribué moderne.

Bénéfices Attendus

- Réduction temps de cycle : 72h → 36h (50%)
- Réduction transferts : 6 → 3 (50%)
- Conformité souveraineté données : 100%
- Scalabilité : illimitée (infrastructure client)
- ROI estimé : 18 mois

Prochaines Étapes

1. Validation de l'architecture par le comité de direction
2. Constitution de l'équipe projet (interne + experts externes)
3. Lancement Phase 1 : développement Agent et API (Mois 1-2)
4. Points de contrôle mensuels avec sponsors (Cathy, Jeanne)