

Design & Development of Eco Business Operating Platform

Central Business Core + Multi-Site Headless Architecture + Guided Selling + CPQ + Workflow

Project Type: Enterprise Business Platform

Client: Lendoux International / Eco Ecosystem

Target: International B2B Trading, Engineering, Integration & E-Commerce

Preferred Development Location: India Development Team

1. Company Background

We are an international engineering and trading company operating across multiple countries.

Our business model combines:

- International trading
- B2B e-commerce
- Engineering consulting
- System integration
- Product selection
- Solution design
- Procurement
- Import/export
- Installation
- Commissioning
- Technical support

We work with approximately:

- **100+ brands**
- **10,000+ products**
- Multiple product categories
- Customers in more than 8 countries
- Thousands of technical and commercial documents
- Approximately 15,000 historical solution/project documents
- Thousands of product relationships, compatibility rules and solution combinations

Our product portfolio includes:

- Data Center equipment
- Servers
- Storage

- Networking
- Industrial networking
- Security and surveillance
- Telecommunications
- Medical equipment
- Energy equipment
- High-tech/OEM products
- Luxury residential technical equipment

Our existing e-commerce platform is based on PrestaShop 8.1, but due to scalability, customization, maintenance, plugin compatibility and automation limitations, we intend to move toward a new architecture.

2. Project Vision

We do **not** want to build another traditional e-commerce website.

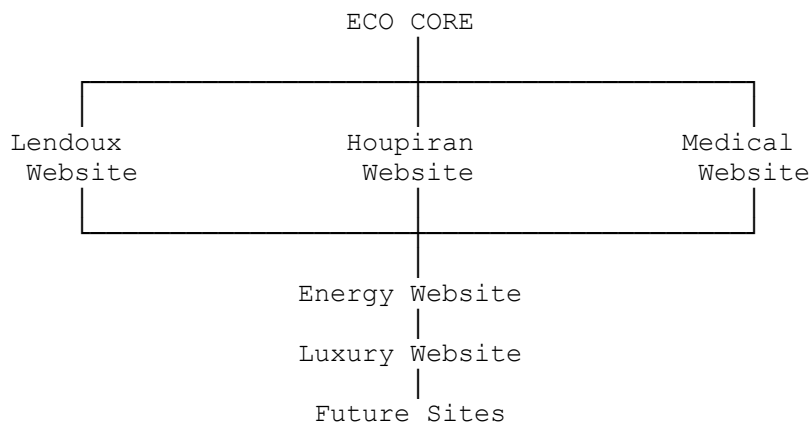
The objective is to build a centralized **Business Operating Platform**, internally called:

Lendoux ECO SYSTEM

Lendoux Eco will become the central business intelligence and operational core of the company.

Websites will act as different customer-facing channels connected to this central core.

The target architecture is:



The central system should control common data, rules, products, solutions, pricing, workflows and business logic.

3. Main Objective

The primary objective is to reduce dependency on highly skilled employees.

A university-educated but non-specialist employee should be able to:

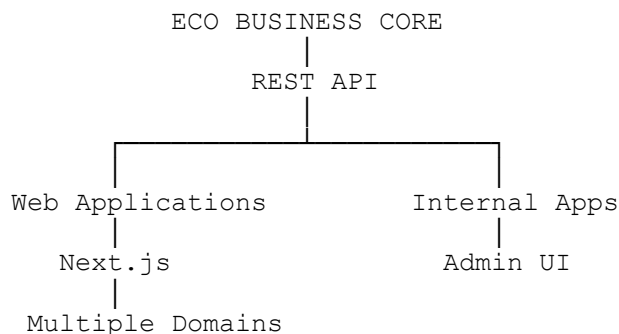
1. Receive a customer request.
2. Identify the customer's requirement.
3. Follow a guided questionnaire.
4. Select the required technical parameters.
5. Receive recommended products/solutions.
6. Generate a standardized quotation.
7. Attach relevant technical documents/videos.
8. Send the proposal to the customer.
9. Automatically route complex cases to technical/commercial staff when required.

The system must therefore operate as a combination of:

PIM + Knowledge Base + Guided Selling + Expert System + CPQ + Workflow + CRM integration + Multi-Site CMS

4. Core Architecture

The preferred architecture is:



The system must be API-first and headless.

The frontend should preferably use:

Next.js / React

Backend may use:

ASP.NET Core or Django

The development company may recommend an alternative architecture if technically justified.

The proposal must clearly explain the recommended technology stack and why.

5. Multi-Site Architecture

The system must support multiple independent websites from a single central platform.

Initial websites may include:

Lendoux

International B2B e-commerce platform.

Houpiran

High-tech/OEM/manufacturer-oriented platform.

Medical Equipment

Dedicated medical equipment platform.

Energy Equipment

Dedicated energy equipment platform.

Luxury Residential

Dedicated luxury technical/residential equipment platform.

Additional websites must be possible without rebuilding the core.

6. Multi-Tenant / Multi-Brand Requirements

Each website must be independently configurable regarding:

- Domain
- Logo
- Brand identity
- Theme
- Navigation
- Language
- Content
- Product visibility
- Categories
- SEO metadata
- Menus
- Landing pages
- Contact information

However, products and business entities should be maintained centrally.

Example:

Product:
Dell Unity 480

Central Core:
Brand = Dell
Category = Storage
Specifications = ...
Documents = ...
Solutions = ...

Website Visibility:

Lendoux = YES
Houpiran = NO
Medical = NO
Energy = NO

7. Product Information Management

The system must support approximately 10,000+ products initially and scale significantly beyond this.

Product structure must support:

- Product
- SKU
- Manufacturer
- Brand
- Category

- Subcategory
- Model
- Technical specifications
- Options
- Variants
- Accessories
- Compatible products
- Alternative products
- Replacement products
- Related products
- Documents
- Videos
- Images
- Solutions
- Industries
- Applications
- Availability
- Lifecycle
- Supplier
- Distributor
- Country/region availability

8. Important Product Lifecycle Requirement

A product must **not simply be deleted** when a supplier relationship ends.

For example:

Product: XYZ Server

Status:

Active

Inactive

Discontinued

Temporarily unavailable

Supplier relationship ended

Region unavailable

Historical

Historical SEO URLs and product information must remain available when appropriate.

9. Guided Selling / Wizard Engine

This is one of the most important modules.

The system must provide a **Wizard Builder** that allows administrators to create guided product-selection workflows without changing source code.

Example:

Customer searches:

Time Server for Instrumentation

or:

Hopf Time Server

or:

8030NTS

The customer enters a guided selection interface.

Question 1

What do you need?

- Time Server
 - NTP Server
 - PTP Grandmaster
 - Other
-

Question 2

Application

- Instrumentation
 - Power Plant
 - Telecom
 - Data Center
 - Industrial Automation
 - Other
-

Question 3

Required output protocol

- NTP
 - PTP
 - Modbus
 - RS232
 - RS485
 - IRIG-B
 - Other
-

Question 4

Oscillator

- TCXO
- OCXO
- Rubidium
- Unknown

If the customer does not know, the system should provide a recommended default.

Question 5

Preferred Brand

- Hopf
 - Meinberg
 - Microchip
 - No preference
-

The system then recommends suitable products.

10. Wizard Must Work for Both Customer and Employee

This is critical.

The same Wizard must be usable by:

Customer

Through the website.

Sales employee

During a telephone conversation.

The employee should be able to open exactly the same Wizard and answer questions on behalf of the customer.

Therefore:

```
Customer
  ↓
Wizard

OR

Phone Call
  ↓
Employee
  ↓
Same Wizard
```

There must not be two different business logic systems.

11. Rule Engine

The system must provide a configurable Rule Engine.

Rules should be stored as data/configuration and **not hard-coded wherever possible**.

Example:

IF

Application = Instrumentation

AND

Protocol = NTP

AND

Oscillator = Rubidium

THEN

Increase Product Score

Another example:

IF

Country = Qatar

AND

Product Category = Storage

THEN

Apply Qatar Logistics Rule

The administrator should be able to create/edit rules without requiring a developer for every change.

12. Product Recommendation Engine

The system should support scoring.

Example:

Product	Score
Hopf 8030NTS	98
Meinberg LANTIME	95
Microchip	90

The scoring model should consider:

- Technical compatibility
- Customer requirements
- Industry
- Budget
- Brand preference
- Availability
- Country

- Supplier
- Lifecycle
- Company preference
- Historical recommendations

The architecture should allow AI-assisted recommendation in the future, but the initial system must remain deterministic and rule-based.

13. Solution Engine

The system must not be limited to individual products.

A customer requirement may produce a **solution consisting of multiple products**.

Example:

Customer Requirement

↓

Time Synchronization System

↓

Recommended Solution

- Time Server
- Antenna
- GPS receiver
- Network equipment
- Software
- Installation

The system should support predefined solution templates.

14. CPQ – Configure, Price, Quote

The system must generate quotations automatically.

Input:

Customer
Country
Products

Quantity
Currency
Discount
Shipping
Customs
Services
Installation

Output:

- Standard quotation
- Proforma invoice
- Commercial offer
- Technical offer
- PDF
- Email

Quotation templates must be configurable.

15. Logistics / Shipping / Customs

The platform should support rules based on:

- Customer country
- Customer city
- Product category
- Weight
- Dimensions
- Incoterm
- Shipping method
- Supplier location
- Destination
- Customs rules
- Local taxes

The architecture must allow integration with external shipping/logistics APIs in the future.

16. Workflow Engine

Different quotations should follow different approval workflows.

Example for a simple product:

Customer Request
↓
Wizard
↓
Quote
↓
Email Customer

Complex project:

Customer Request
↓
Sales
↓
Technical Review
↓
Commercial Review
↓
Management Approval
↓
Quote
↓
Customer

Workflow must be configurable.

17. Document / Knowledge Management

We currently have approximately **15,000 historical solution and project documents**.

The system must support:

- PDF
- Word
- Excel
- Images
- Videos
- Datasheets
- Presentations
- Technical documents
- Installation manuals
- Project documentation
- Solution documents

Documents should be associated with:

- Products

- Brands
 - Solutions
 - Industries
 - Projects
 - Applications
 - Customers
-

18. Knowledge Relationships

The system must support relationships such as:

```
Product
  ↓
Brand
  ↓
Application
  ↓
Industry
  ↓
Solution
  ↓
Project
  ↓
Document
  ↓
Video
```

and:

```
Product A
  ↓
Compatible with
  ↓
Product B
Product A
  ↓
Alternative
  ↓
Product B
Product A
  ↓
Replacement
  ↓
Product B
```

19. CRM / Customer Inquiry

The platform should capture:

- Customer
- Contact
- Country
- Industry
- Request
- Products
- Requirements
- Wizard answers
- Recommended solution
- Quote
- Communication history
- Status
- Assigned employee

It must be possible to integrate with external CRM systems later.

20. Website Search

The websites must support advanced search.

Customer may search:

Dell Unity 480

or:

Time Server

or:

8030NTS

or:

DIN Rail Time Server

Search should support:

- Product
- Brand
- SKU
- Model

- Technical specification
 - Application
 - Industry
 - Solution
-

21. SEO

SEO is extremely important because a major source of customer acquisition is Google.

The system must support:

- SEO-friendly URLs
- Metadata
- Canonical URLs
- Structured data/schema
- XML sitemap
- Robots management
- Hreflang
- Multilingual SEO
- Redirect management
- Product structured data
- Category structured data
- Historical URL preservation
- Server-side rendering / static generation where appropriate

Existing SEO URLs should be preserved or redirected during migration.

22. Languages

Initial Lendoux website supports:

/fa/
/en/
/tr/
/ar/

The architecture must support additional languages without architectural changes.

23. Headless Frontend

Preferred technology:

Next.js / React

The frontend should obtain business information through secure APIs.

The frontend should be:

- Fast
 - SEO-friendly
 - Responsive
 - Mobile-friendly
 - Accessible
 - Secure
 - CDN-compatible
-

24. Central Administration Panel

The administration system is one of the most important components.

Administrators must be able to manage:

- Products
- Brands
- Categories
- Specifications
- Documents
- Solutions
- Rules
- Wizards
- Questions
- Answers
- Workflows
- Prices
- Customers
- Quotations
- Websites
- Languages
- SEO
- Users

- Permissions
-

25. Role-Based Access Control

Minimum roles:

Super Admin

Product Manager

Sales

Technical

Commercial

Purchasing

Logistics

Website Content Manager

Management

Each role must have configurable permissions.

26. API Architecture

The Core must provide APIs for:

- Websites
- Mobile applications in the future
- CRM
- Marketplaces
- ERP
- Logistics providers
- Payment systems
- AI services
- External integrations

Existing Lendoux APIs for marketplaces such as Torob and Digikala must be considered during migration.

27. Search Engine

The architecture should support a dedicated search engine such as:

- Elasticsearch
- OpenSearch
- Meilisearch

The developer should recommend the most appropriate option.

Search must scale beyond 10,000 products.

28. Infrastructure

The initial objective is to operate multiple websites from a centralized infrastructure.

Preferred architecture should support:

- Linux
- Docker
- Reverse proxy
- SSL
- CDN
- Database
- Redis/cache
- Object storage
- Search
- Automated deployment
- Monitoring
- Backup

The architecture must allow individual services to move to separate servers later.

29. Security

The system must include:

- MFA
- Role-based access
- API authentication
- Rate limiting
- Input validation
- Audit logs
- Encryption
- Secure secrets management
- Database security
- Backup encryption
- Firewall
- Security headers
- Dependency management
- Vulnerability monitoring

The proposal must explain the security architecture.

30. Backup & Disaster Recovery

The developer must provide a backup strategy for:

- Database
- Documents
- Images
- Configuration
- Source code

The system must support restoration testing.

31. Migration from Existing Systems

Existing systems include:

- PrestaShop
- Excel databases
- Product databases
- Existing documents
- Existing URLs
- Existing product information
- Existing APIs

- Existing SEO structure

The developer must provide an import/migration strategy.

We do NOT want to manually re-enter thousands of products and documents.

CSV/Excel/API/bulk import tools are required.

32. Important Development Principle

The following items must be configurable through the administration panel wherever practical:

- Questions
- Answers
- Wizard paths
- Rules
- Product scores
- Workflows
- Email templates
- Quote templates
- Document associations
- Website visibility
- Product relationships

These should not require source-code changes for normal business modifications.

33. AI – Future Phase

AI should NOT be the foundation of the initial system.

However, the architecture must be AI-ready.

Future capabilities may include:

- AI product recommendation
- Natural language product search
- AI sales assistant
- Customer requirement extraction
- Technical document search
- Solution recommendation

- Automatic response generation

AI must operate on top of structured company knowledge and business rules.

34. Development Methodology

We do NOT want a six-month "big bang" development project.

Development must be incremental.

The first production pilot should be:

TIME SERVER / TIME SYNCHRONIZATION

Example brands:

- Hopf
- Meinberg
- Microchip

Example use cases:

- Instrumentation
- Telecom
- Data Center
- Power
- Industrial

This module should demonstrate:

```
Search
↓
Wizard
↓
Rules
↓
Recommendation
↓
Product
↓
Pricing
↓
Quote
```

↓
Approval
↓
Email

Once this workflow works in production, the architecture will be expanded to other product families.

35. Required Deliverables from Developer

The proposal must include:

A. Architecture

Complete system architecture diagram.

B. Technology Stack

Explain:

- Backend
- Frontend
- Database
- Search
- Cache
- Workflow
- Authentication
- Infrastructure

C. UI/UX

Admin dashboard + customer website + employee sales interface.

D. Source Code

Full source code ownership.

E. Documentation

Technical documentation and API documentation.

F. Deployment

Production deployment.

G. Testing

Unit testing, integration testing and security testing.

H. Training

Training for our internal team.

I. Maintenance

Optional annual maintenance proposal.

36. Required Pricing Structure

Please **do not provide only one total project price.**

Provide separate prices for:

Module	Development Cost
Architecture	
Core Backend	
Database	
Admin Panel	
PIM	
Knowledge Management	
Wizard Builder	
Rule Engine	
Recommendation Engine	
Solution Engine	
CPQ / Quotation	
Workflow	
CRM/Inquiry	
Document Management	
Search	
API	
Next.js Frontend	
Lendoux Migration	

Module	Development Cost
Houpiran Website	
Multi-site architecture	
SEO migration	
Infrastructure / DevOps	
Security	
Testing	
Documentation	
Training	
Annual Support	

Also provide:

- Hourly rate
- Daily rate
- Monthly team cost
- Estimated man-hours
- Estimated timeline
- Team members required

37. Team Requirements

Please specify proposed team:

- Solution Architect
- Backend Developer
- Frontend Developer
- UI/UX Designer
- DevOps Engineer
- QA Engineer
- Database Engineer
- Project Manager

For each person specify:

- Experience
- Years
- Technology
- Monthly cost
- Allocation %

38. Ownership

All:

- Source code
- Database schema
- Documentation
- APIs
- Configuration
- Business rules
- UI designs

developed specifically for this project must belong to the client.

No vendor lock-in is acceptable.

39. Important Commercial Requirement

The developer must clearly separate:

One-time development

from:

Recurring costs

including:

- Hosting
 - Cloud
 - CDN
 - Third-party APIs
 - Licenses
 - Monitoring
 - Email
 - Search
 - Backup
 - Support
-

40. Expected Proposal Format

Please respond with:

1. Understanding of the project
2. Proposed architecture
3. Technology stack
4. Architecture diagram
5. Development methodology
6. Project phases
7. Timeline
8. Team
9. Cost by module
10. Hourly/day/monthly rates
11. Infrastructure cost
12. Third-party license costs
13. Migration strategy
14. Security strategy
15. Backup/DR strategy
16. Maintenance cost
17. Warranty period
18. Similar projects
19. References
20. Risks and assumptions

41. Critical Requirement

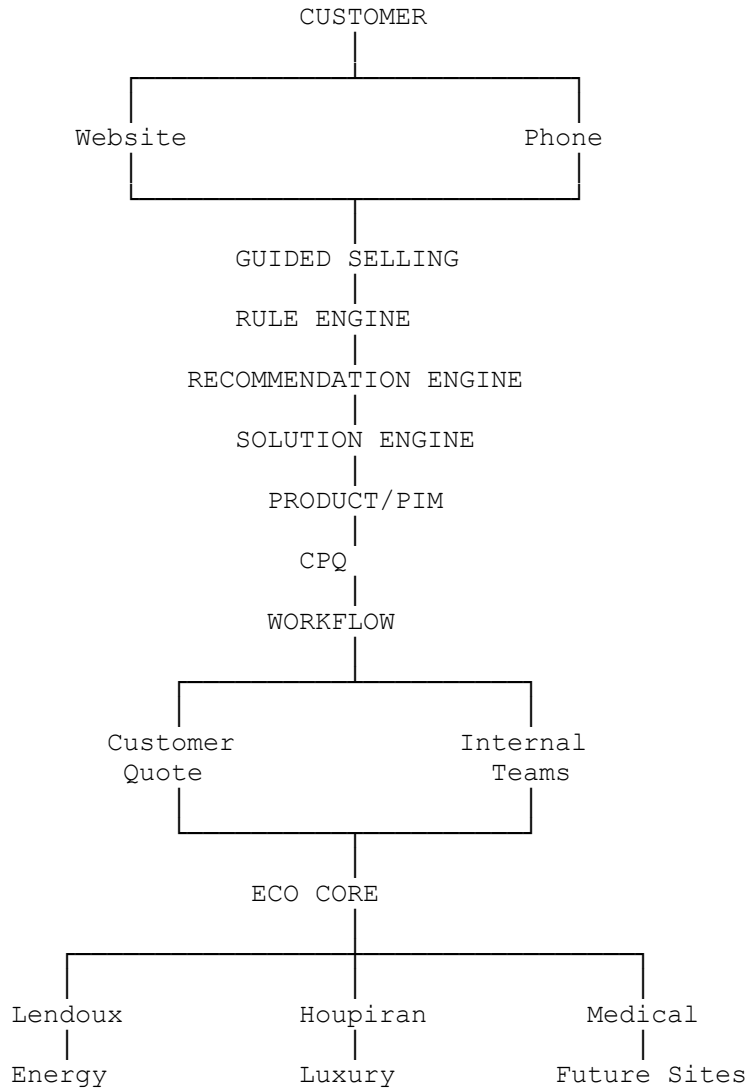
The vendor must understand that this is **NOT primarily an e-commerce website project**.

The central objective is to build:

A centralized business operating platform capable of managing products, brands, solutions, technical knowledge, guided selling, rules, quotations and workflows, while powering multiple independent websites.

The websites are channels connected to the central business core.

42. Target Architecture Summary



43. First Pilot

The first production pilot should be **Time Synchronization / Time Server**.

The vendor should demonstrate:

Scenario A – Website

Customer searches:

Time Server for Instrumentation

→ Guided Wizard

→ Questions

- Recommendation
- Product
- Price
- Shipping
- Quote
- PDF
- Email

Scenario B – Telephone

Sales employee receives a phone call.

Employee opens the same Wizard.

Employee answers the questions based on the customer's answers.

The exact same Rule Engine generates the recommendation.

This is the minimum demonstration required before expanding the system.

44. Final Note to Vendor

We are looking for a **long-term technology partner**, not simply a team to develop a website.

The system will evolve continuously because:

- Products change.
- Brands change.
- Suppliers change.
- Prices change.
- Countries change.
- Rules change.
- Services change.
- New websites are added.
- New industries are added.

Therefore, the architecture must be **modular, configurable, API-first, scalable and maintainable for 10+ years**.