Apps that humans and robots love: optimize your websites for AI agents

ibexa

Hey. I'm G/

Love to tinker with stuff.
Code, cameras, watches.

Love buying domain names for unfinished side projects.

Started PHP in 98.
Started using SF1 in 2003.
Started using "Ibexa" in 2010.

"Vibe coder" since 2023. Or not.

ibexa

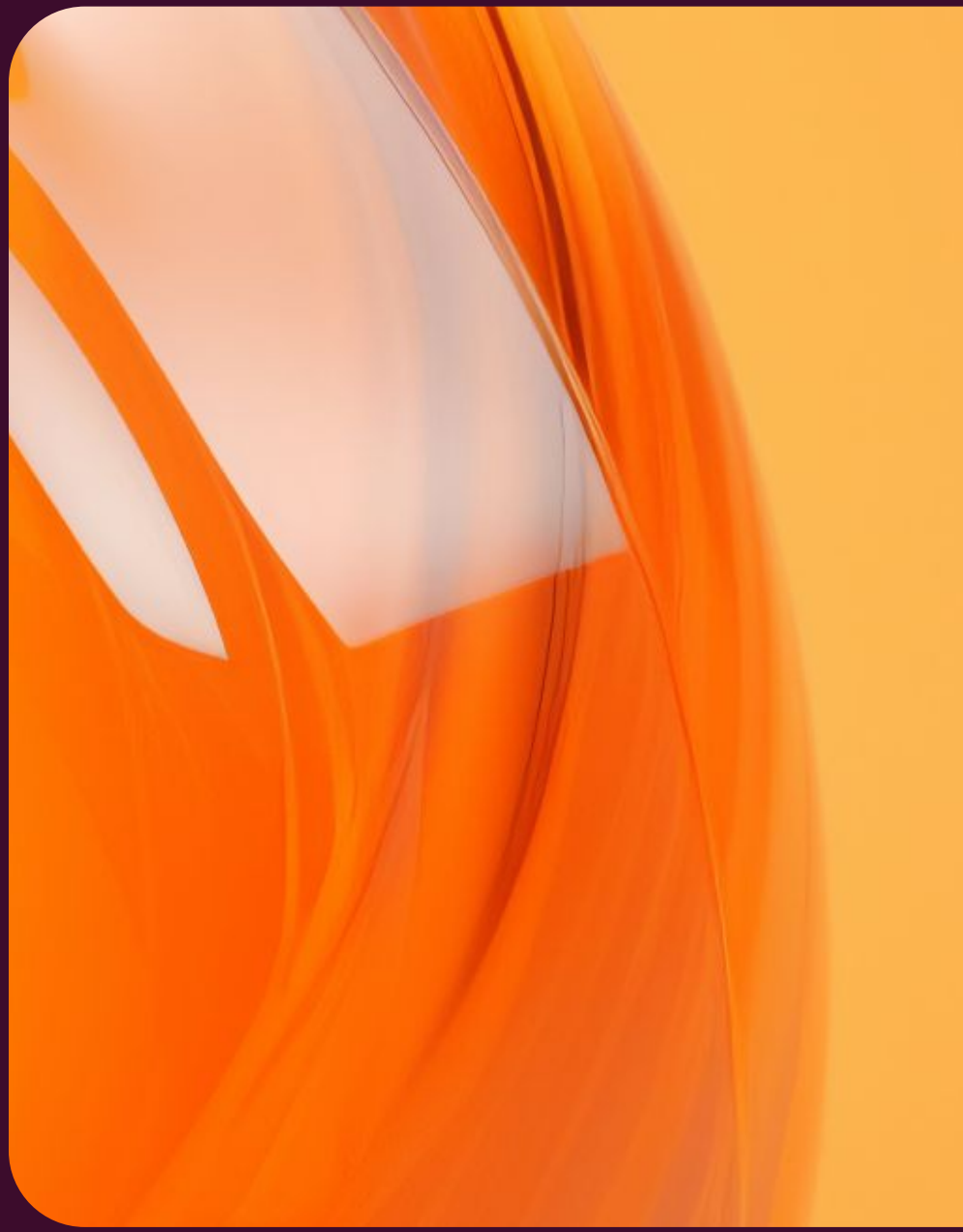
The Silent Revolution

ibexa

Last month, an AI agent...

Booked a restaurant
Compared insurance quotes
Filed a support ticket...

all by browsing websites.



ibexa

Was your site one of them?

ibexa

“51% of internet traffic is now automated”

— Imperva Bad Bot Report 2025

ibexa
Summit 2026

+1,300%

AI traffic to retail sites (year-over-year growth)

The new intermediary


Meet Your
New Users
AI Agents Explained
ibexa

Types of AI agents browsing the web

1. Conversational AI with browsing: ChatGPT, Claude, Gemini
Find me a hotel in Paris for next weekend

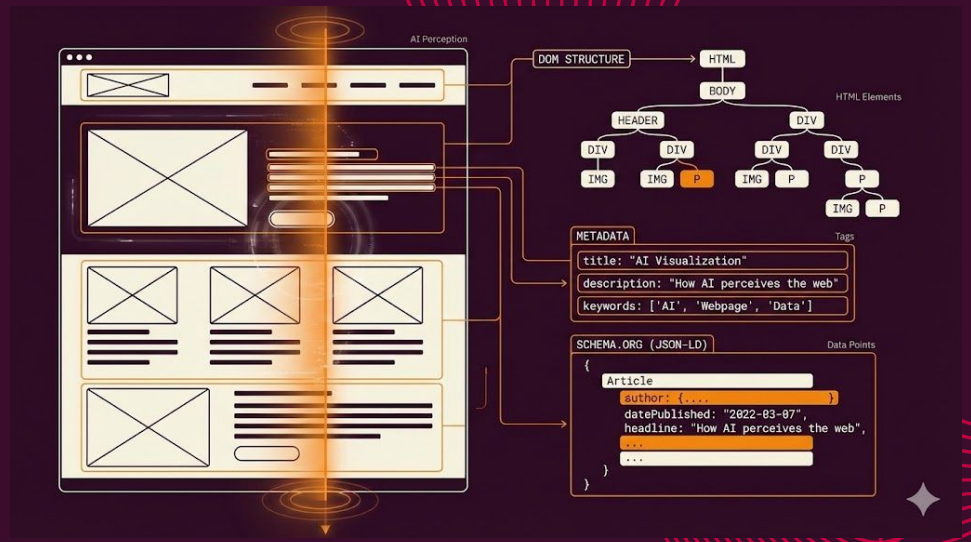
2. Research & discovery engines: Perplexity, You.com, Phind
Synthesize information from multiple sources

3. Automation agents: Browser-use tools, Enterprise RPA
Complete multi-step tasks autonomously

4. Specialized crawlers: AI-powered SEO tools, content aggregators
Index and analyze at scale

ibexa

How agents "see" your website



ibexa

What agents extract

Layer	What They Read
Text	Headings, paragraphs, lists
Structure	DOM hierarchy, semantic elements
Metadata	Schema.org, Open Graph, meta tags
APIs	REST endpoints, GraphQL
Visual	Screenshots (emerging)

ibexa

Where they struggle

Layer	The issue
JavaScript-heavy SPAs	No server-side rendering = invisible content
Content behind auth walls	Agents can’t log in (usually)
Div soup	No semantic structure = no understanding
Ambiguous navigation	"Click here" tells them nothing
Missing or conflicting metadata	Schema.org says €99, page says €149

The Beautiful Convergence

Optimizing for AI agents is not another separate task.
It's an acceleration of best practices you should already be doing.

ibexa

Accessibility helps agents

A11y Practice	Agent Benefit
Semantic HTML	Structure understanding
ARIA labels	Element purpose
Alt text	Image comprehension
Clear navigation	Path discovery

ibexa

SEO helps agents

SEO Practice	Agent Benefit
Structured data	Machine-readable content
Meta descriptions	Quick summaries
Heading hierarchy	Content organization
Performance	Fast responses

ibexa

Why Ibexa partners have an advantage

Structured content modeling already separates content from presentation

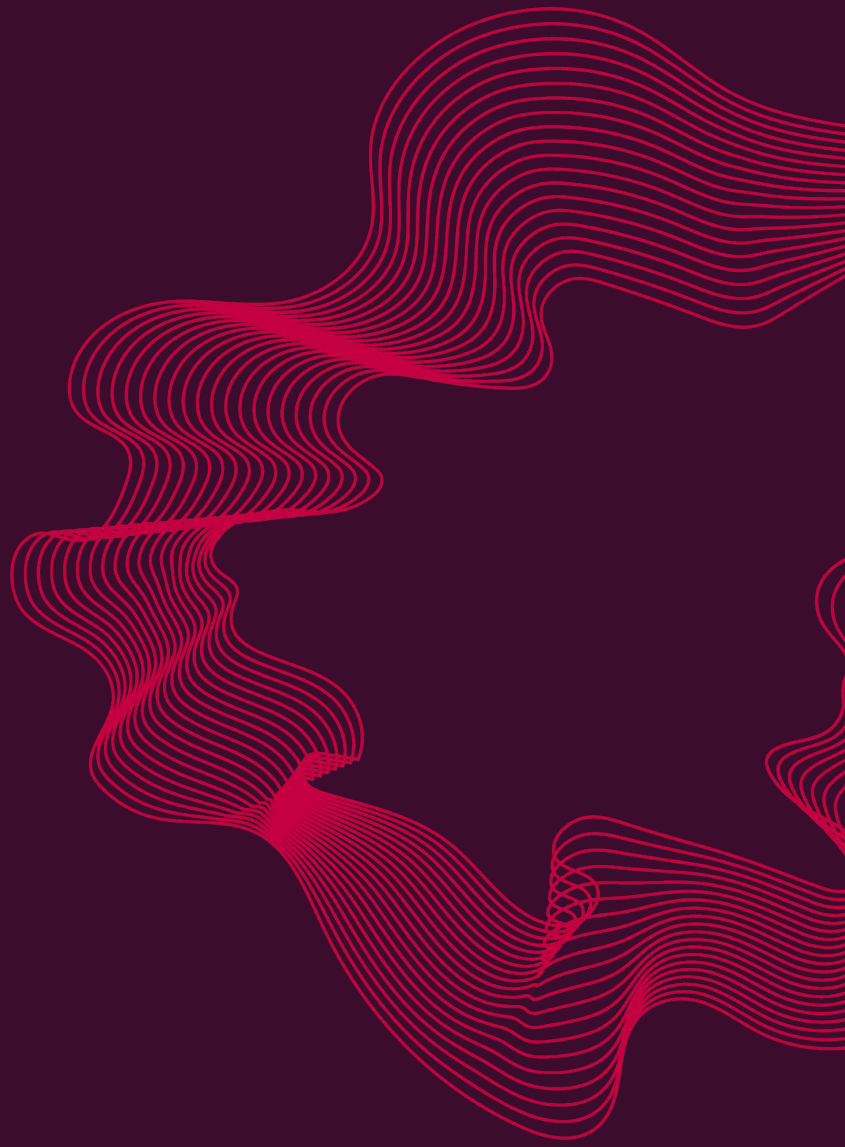
Content types with defined fields = machine-understandable content

Headless/hybrid approaches enable multiple output formats

ibexa

Web Optimization for AI Agents

From easiest to most advanced



ibexa

The complexity ladder

Level	Technique	Effort
★	Content clarity	No code
★★	Semantic HTML	Templates
★★★	Schema.org	Integration
★★★★	llms.txt	New standard

ibexa


⭐ Content Clarity
No code changes required

Content clarity checklist

Before touching any code, improve your content.

Clear writing for humans is clear data for agents.

- Clear, unambiguous language
- Summaries at the top of long pages
- Explicit definitions for jargon
- FAQ sections
- Consistent terminology

❌ *"Our revolutionary solution leverages cutting-edge technology..."*

✅ *"This software automates invoice processing. It costs €99/month and integrates with SAP, Oracle, and QuickBooks."*

ibexa

⭐⭐ Semantic HTML + ally

Template changes only

ibexa

Everything that helps screen readers helps AI agents.

WCAG compliance = agent-friendly

If your HTML is a mess of s, agents struggle to understand page structure.

ibexa

Semantic elements matter

```
<!-- Bad: Div soup -->
<div class="product-box">
  <div class="title">Widget Pro</div>
  <div class="btn" onclick="buy()">Click here</div>
</div>

<!-- Good: Semantic structure -->
<article itemscope itemtype="https://schema.org/Product">
  <h2 itemprop="name">Widget Pro</h2>
  <button aria-label="Add Widget Pro to cart">
    Add to Cart
  </button>
</article>
```

ibexa

Schema.org

The Rosetta Stone

ibexa

JSON-LD example

```json
{
  "@context": "https://schema.org",
  "@type": "Product",
  "name": "Widget Pro",
  "offers": {
    "@type": "Offer",
    "price": "99.99",
    "priceCurrency": "EUR",
    "availability": "https://schema.org/InStock"
  }
}
```

ibexa

Priority order

Organization — who you are

Product/Service — what you sell

BreadcrumbList — site structure

FAQPage — common questions

Article — for content sites

llms.txt

The emerging standard

ibexa

Example llms.txt

llms.txt tells AI agents HOW to understand and use your site (not what to block)

```
# Company: Acme Corp
# Purpose: E-commerce for industrial widgets

## Key Pages
- /products - Browse all products
- /api/docs - REST API documentation

## Content Freshness
- Product prices updated daily at 6:00 UTC
- Inventory levels are real-time
```

ibexa

Why adopt now?

Minimal investment — one text file

Early adoption — signals forward-thinking

Future-proof — the practice will persist even if format evolves

API Design for AI Agents

The technical deep-dive

ibexa

The context poverty problem

Traditional REST APIs were designed for human developers reading documentation.

AI agents need to discover, understand, and use your APIs autonomously.

```
Agent must call:
  GET /customerProfile
  GET /accountStatus
  GET /paymentHistory
  GET /offerEligibilityRules

Then synthesize into a single decision
```

ibexa

The consequences

Context bloat increases 3-5x

Error rates climb

Agent reasoning becomes brittle

80% of enterprise APIs lack agent optimization

⭐⭐ Structured errors

Modify existing error handlers

ibexa

Bad error response

```
{
  "error": "Something went wrong"
}
```

Agent has no idea what to do next. Do this instead:

```
{
  "error": {
    "code": "PRODUCT_NOT_FOUND",
    "message": "Product 'widget-xyz' does not exist",
    "retryable": false,
    "suggestions": {
      "similar_products": "/products/search?q=widget"
    }
  }
}
```

Unified endpoints

New endpoint design

ibexa

Fragmented vs Unified

```
GET /products/{id}
GET /inventory/{productId}
GET /pricing/{productId}
GET /shipping/{productId}
GET /reviews/{productId}

5 round trips, 15KB+ data
```

```
GET /products/{productId}/purchase-context

{
  "product": { "name": "Widget Pro", "price": 99.99 },
  "shipping": { "estimatedDelivery": "2026-02-07" },
  "reviews": { "average": 4.7, "count": 324 }
}

1 call, 2KB data
```

Optimization techniques

Technique	Benefit
Field selection	Reduce context 50-70%
Batch operations	Reduce calls by 90%
Compression	Reduce payload 70-90%
Cursor pagination	O(1) performance

OpenAPI + HATEOAS

Enhance existing specs

ibexa

Semantic OpenAPI extensions

```yaml
paths:
  /products/{id}/availability:
    get:
      x-action: "CHECK_AVAILABILITY"
      x-intent: "Determine if product can ship by date"
      x-category: "inventory-management"

# Agents understand WHEN to use this endpoint
```

ibexa

HATEOAS: Guided navigation

```
{
  "productId": "12345",
  "stock": 47,
  "links": {
    "add-to-cart": "/cart/add?product=12345",
    "check-delivery": "/products/12345/delivery-options"
  }
}

# Agents discover next steps dynamically
```

ibexa

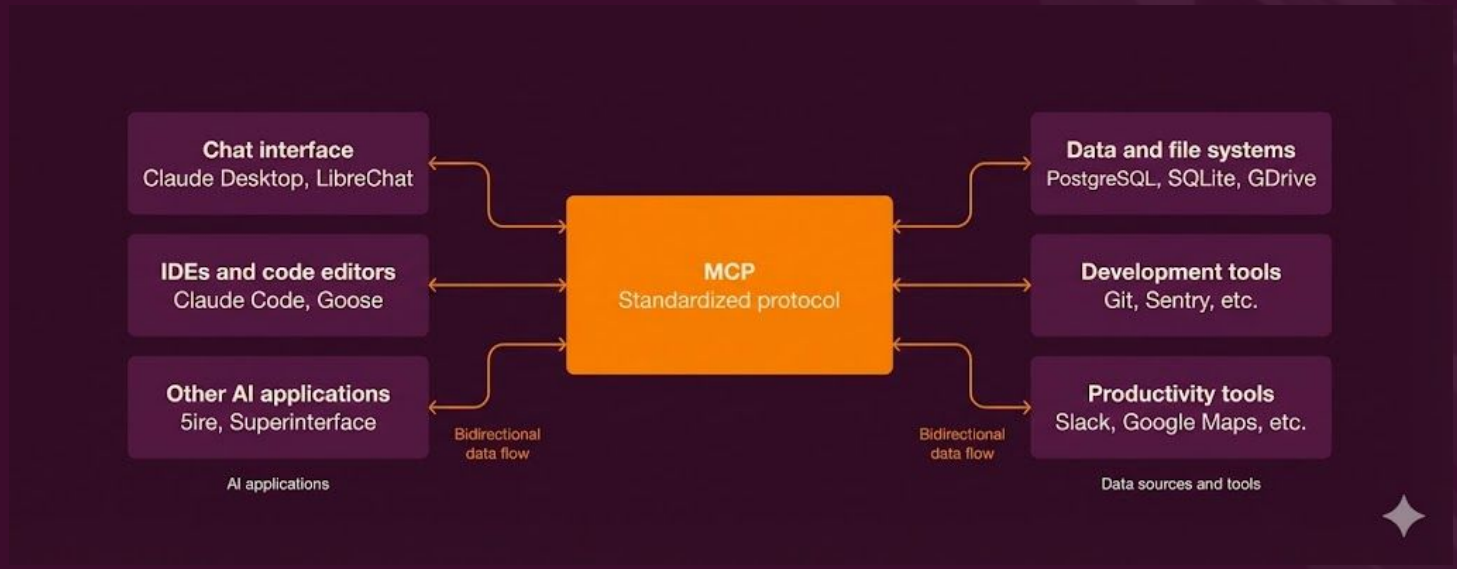
Model Context Protocol

The emerging standard

ibexa

What is MCP?

Think of MCP like a USB-C port for AI applications. Just as USB-C provides a standardized way to connect electronic devices, MCP provides a standardized way to connect AI applications to external systems.



Tools — Functions agents can call. Resources — Data sources to query. Prompts — Guidance for when/how to use tools

ibexa

The honest downsides

Challenge	Reality
Separate product	Additional infrastructure
Context overload	50+ tools exhaust context
Duplication	Logic in two places
Immature ecosystem	Build custom tooling

Ibexa DXP

Your AI-Ready Foundation

ibexa

Why you have an advantage

Structured Content Model — Consistency enforced

Semantic Field Types — Rich text, relations, taxonomies

Multi-channel Output — Same content, different formats

API-First — REST and GraphQL built-in

Personalization Engine — Adapt for different "users"

Ibexa API advantages

REST API — Already structured

GraphQL — Flexible queries

Content types — Natural API schemas

Emerging Standards

The future is being written now

ibexa

The “standards” landscape

Standard	Focus	Priority
Schema.org	Structured data	Now
llms.txt	AI guidance	Now
MCP	LLM-to-tool	Q2
UCP	E-commerce	Q1-Q2
NLWeb	Conversational	Q2-Q3
A2A/AP2	Negotiation	2027

ibexa

Universal Commerce Protocol (UCP)

Google + Shopify + Others

One protocol for all agent-store interactions

```
Agent discovers store →
Reads /.well-known/ucp manifest →
Queries products → Creates cart →  Completes checkout

# All via standardized protocol
```

ibexa

UCP capabilities

Capability	What it enables
Discovery	Agents find your store
Querying	Natural language → products
Transactions	Cart, checkout, payment
Post-Purchase	Tracking, returns, support

ibexa

Natural Language Web (NLWeb)

Conversational queries against your data

```
User: "Leather handbags under €150"
     ↓
Agent → Your NLWeb endpoint
     ↓
Your system parses, queries, returns JSON
     ↓
Agent: "I found 2 options..."
```

ibexa

Testing with AI Agents

Validate your optimizations

ibexa
Summit 2026

You wouldn't launch without testing in a browser.

Now you need to test with an AI agent too.

ibexa

Manual testing: Ask the agent

Open ChatGPT, Claude, or Gemini with browsing enabled

Then ask questions about your site (specify the URL).

"What do you understand about this page?"
Validates: Structure, Schema.org, content clarity

"Find product X on this website"
Validates: Navigation, search, product schema

"What is the price and availability of X?"
Validates: Structured data accuracy

"How would I contact support?"
Validates: Navigation clarity, CTA visibility

Automated testing: Ask the agent

Claude Code (or others) agents can also automate this. And even use chromium to browse the site (puppeteer, agent-browser, etc.).

You can even create the testing plan with a coding agent.

Can be triggered on every deploy on the preview and production environments.

Track comprehension reports over time.

Call to Action

What to do Monday morning

ibexa

Key takeaways

You start to think about AI Agent Experience (AIX).
Research and follow the news…

AI agents are real users — treat them as such
Optimization amplifies accessibility and SEO
Structured content is the foundation

Start simple: semantic HTML → Schema.org → llms.txt
New standards are emerging — watch and prepare

ibexa

In the age of AI, the websites that win will be those that are as clear to machines as they are compelling to humans.

Fortunately, these goals are the same.

ibexa

Agent-to-Agent (A2A)

```
Coming: Agents negotiating with agents

Consumer Agent: "€150 + free shipping?"
Retailer Agent: "Accepted"

No human intervention
```

ibexa

Thank you

Questions?

ibexa