

Chunks vs. Bits

How structured content lets AI systems truly understand it

The Challenge

Publishers increasingly face the question of how to future-proof their content for AI-powered applications – for chatbots, retrieval systems (RAG), and interactive products. The common technical answer is: chunking. Documents are broken into small text segments so AI models can search and process them efficiently.

Chunking typically happens after the fact. This method works – but it has a structural weakness: the chunk knows nothing about itself. bitmark takes a different approach: content is structured from the very start into meaningful, typed units – bits.

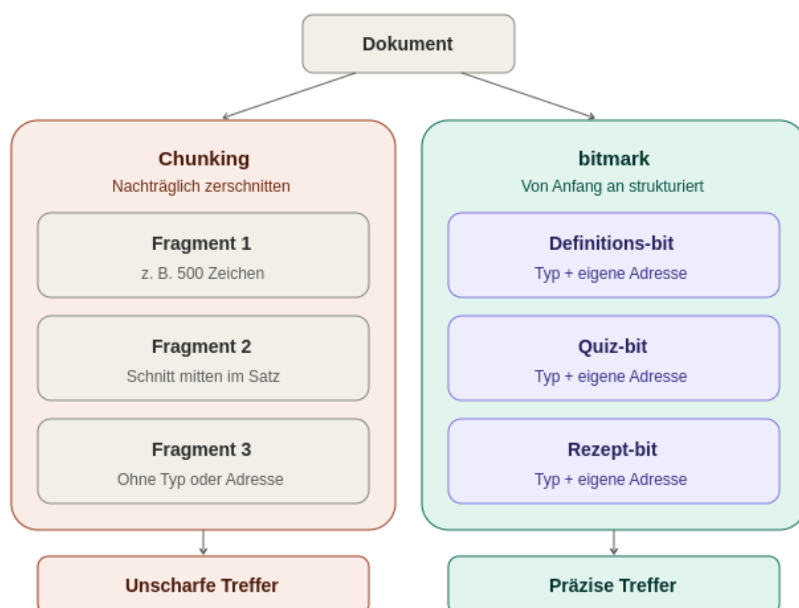
Retroactive Chunks vs. Bits – A Comparison

Criterion	Chunks (classic AI chunking)	Bits (bitmark)
Citability	Traceability varies widely – from raw chunk fragments to, at best, a page reference; the link to the exact source location is usually missing.	Every bit carries a fixed, unique address and a reference to the exact source location – answers are traceable down to the sentence.
Control & independence	Processing often happens as a black box at the system vendor; the customer loses insight into and control over their own knowledge base (vendor lock-in).	bitmark is an open standard. The bits belong to the publisher and are vendor-independently portable between any AI systems.
Creation	Retroactive: a document is technically cut by character or token length.	Created from the outset as a self-contained, meaningful unit.
Semantics	Arbitrary boundary – a chunk can end mid-sentence or mid-thought.	Every bit is a self-contained, independently understandable unit of knowledge.
Structure	Plain text fragment, usually without information about type or function.	Typed object (definition, quiz, recipe ...) – over 80 bit types, some interactive.

Criterion	Chunks (classic AI chunking)	Bits (bitmark)
Reusability	Practically none – the chunk only exists in the context of its indexing.	Freely combinable, referenceable, and versionable as an atomic unit.
Link to the original	Usually lost or only loosely retained (e.g. page number, character offset).	Contains references to the original content – RAG systems can display source and context.
Machine-readability	Readable by language models, but without embedded semantics.	Language models generate bitmark markup today as reliably as JSON.

Visualization

A classic chunking process cuts a document along arbitrary length boundaries. bitmark, by contrast, structures the same content from the outset into self-contained, typed bits with a fixed address – the result is more precise matches in downstream AI applications.



Full Citability: From Chunk to Exact Location

How citable an AI system truly is varies widely in practice. Roughly four levels of maturity can be observed:

- Simple systems merely display the chunks an answer was generated from – loose text fragments with no context.

-
- Better systems point to the entire source document, e.g. a 200-page PDF – the user has to find the relevant spot themselves.
 - Good systems narrow the reference down to individual pages.
 - Professional systems point exactly to the underlying location in the text.

bitmark sits at the fourth level by design: since every bit carries its own fixed address and knows its relation to the original content, the exact location isn't the result of elaborate after-the-fact mapping – it's a structural property of the content itself. For publishers, this is more than a technical detail: it's the foundation for an AI answer actually being reliable – rather than, as it's sometimes put pointedly in practice, a well-phrased claim without evidence.

Control Instead of a Black Box

Many RAG systems today work on a simple but, for publishers, risky pattern: the customer uploads their documents, and the vendor generates the chunks internally – usually proprietary and not inspectable. These chunks stay with the vendor; the customer only receives the finished answers. The result is a classic vendor lock-in: limited negotiating power, high switching costs, and full dependence on a single vendor's roadmap.

bitmark reverses this relationship. Because bitmark is an open, documented standard, the structured content belongs to the publisher throughout – not to whichever AI vendor is in use. Bits can be used independently of the language model deployed, migrated between systems, and carried over to any new vendor without having to reprocess the content. For publishers, this means genuine data sovereignty: the choice of AI vendor stays changeable at any time, without losing the value of the content itself.

Why This Matters for Publishers

- Quality over randomness: bits give AI systems self-contained, meaningful units instead of arbitrarily cut text fragments – noticeably improving match quality in RAG applications.
- Full citability: since every bit carries a fixed address and a link to the original content, any AI answer can be traced back to the exact passage of text – a key point for publishers with a duty of care toward authors and licensors.
- Control instead of a black box: the bits belong to the publisher, not the AI vendor – protecting against vendor lock-in and preserving negotiating power with changing technology partners.
- Reusability: bits can be combined into new products independently of the source document – textbooks, reference works, interactive applications – without reprocessing the content.

-
- Future-proofing: bitmark is an open standard that has proven stable for three years, even as more than 80 new bit types have been added – investments in bitmark content remain usable long-term.
 - AI compatibility by design: language models already generate bitmark markup reliably today – new content can be produced directly in structured form instead of being chunked after the fact.
 - Images, graphics, and tables are made accessible to AI this way as well.
 - Chunks (bits) can be displayed in their original context or standalone, thanks to the associated streaming reader.

How Are Bits Created?

A very practical way to create bits is to use the RRP365 platform from Reemers Publishing Services. With or outside this platform, content is edited and structured in the classic way using Microsoft Word. From this structured data, pre-built templates can then generate print-ready PDFs as well as XML, HTML, EPUB, or bitmark itself.

Conclusion

Retroactive chunking is a technical workaround for unstructured content – with two blind spots: traceability back to the source, and control over where and how one's own content is processed. bitmark makes this workaround unnecessary, because the content exists from the outset in meaningful, typed, addressable, and therefore fully citable units – owned by the publisher, independent of the chosen AI vendor. For publishers, this means: higher content quality for AI applications, reliable provenance, genuine data sovereignty, and a structuring effort that doesn't need to be repeated with every new AI generation.