Information Fractal Is

Information Fractal Is

Information Fractal Is

Information Fractal Is

Information Fractal Is

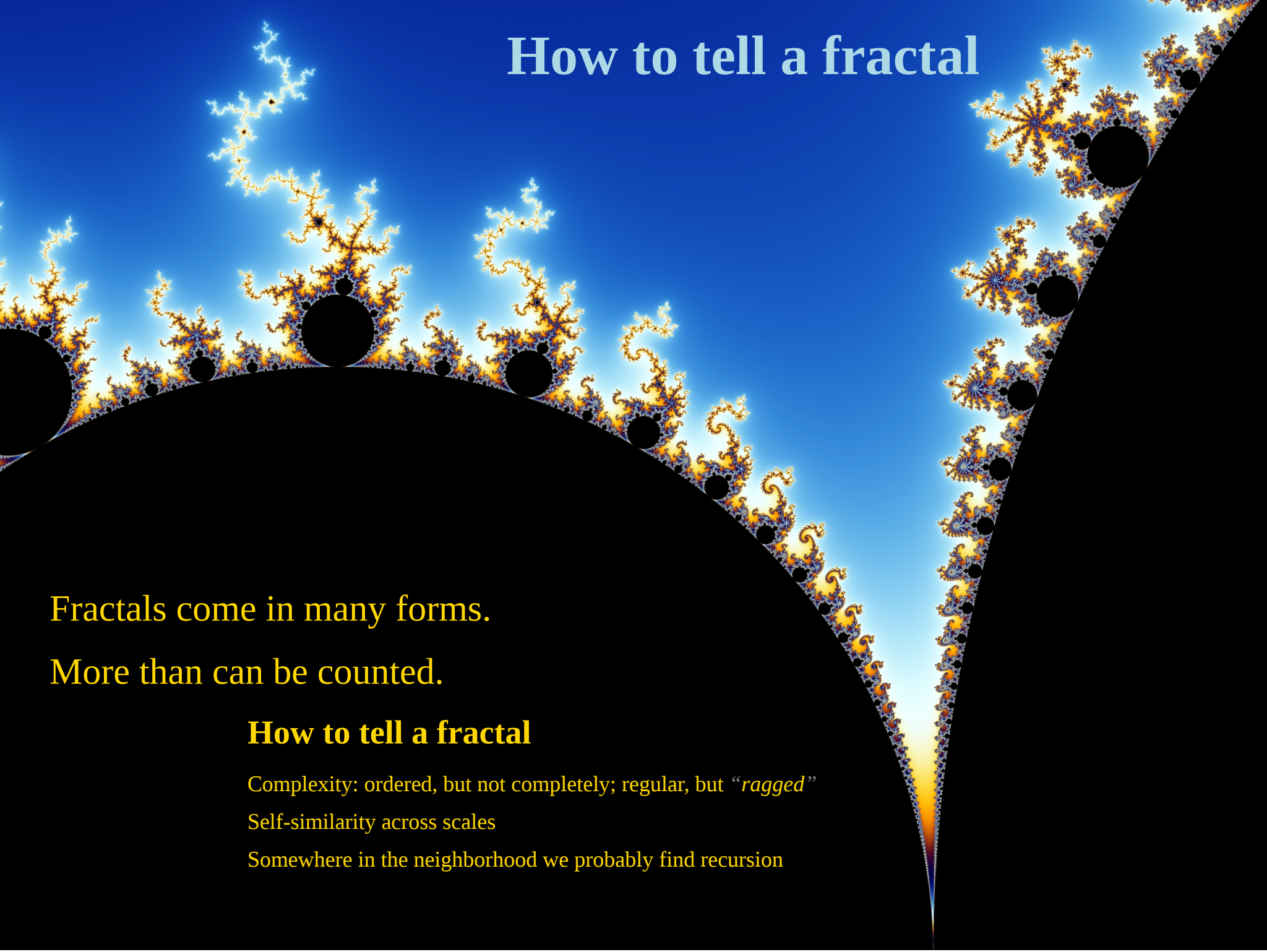
Information Fractal Is

Wendell Piez

Balisage 2018

Rockville, Maryland

How to tell a fractal



Fractals come in many forms.

More than can be counted.

How to tell a fractal

Complexity: ordered, but not completely; regular, but *“ragged”*

Self-similarity across scales

Somewhere in the neighborhood we probably find recursion

We *conceive* fractals with mathematics

We *observe* fractals in nature

Cultural productions also exhibit fractal features

... whether by *“accident”* or *“design”*

This matters to us because ...

It's all about regularity

Or (contrary-wise) scope definition and exception handling

“How granular is your information?” is like *“How long is your coastline?”*

Spoiler alert: with *control* comes *dependency*

Cultural production - the archive!

Documentary production (or: the written word)

Electronic/documentary media

Non-proprietary, open, standards-based media

Text-based formats

Formalisms, formal languages, programming languages

Markup languages and data description syntaxes

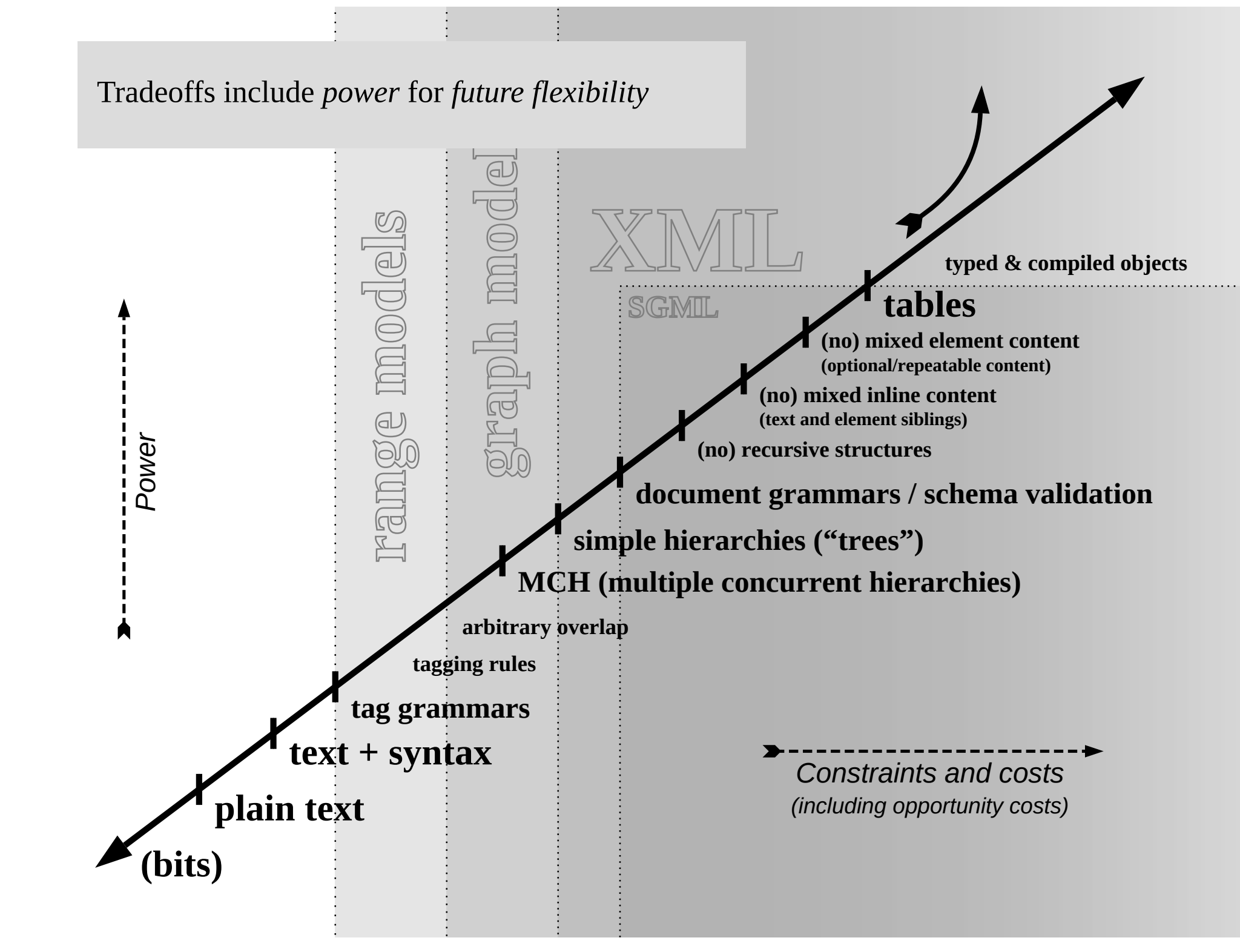
Why is this hard?

If transformational linguistics were correct, this would be no problem.

Instead, we have the semantic soup of human language and culture

(To say nothing of whale-, crow- or elephant-speak)

Somehow, we manage to make sense out of the mishmash is everywhere!


Tradeoffs include *power* for *future flexibility*
range models
graph model
XML
SGML
typed & compiled objects
tables
(no) mixed element content
(optional/repeatable content)
(no) mixed inline content
(text and element siblings)
(no) recursive structures
document grammars / schema validation
simple hierarchies (“trees”)
MCH (multiple concurrent hierarchies)
arbitrary overlap
tagging rules
tag grammars
text + syntax
plain text
(bits)
Power
Constraints and costs
(including opportunity costs)

```
Example of plain text
```

```
Example of *plain text*
```

```
<p>Example of <em>plain text</em></p>
```

```
Example of }{\i\ai\rtlch \ltrch\loch
plain text}
\par }
```

```
{
   "p": {
      "em": "plain text",
      "__text": "Example of"
   }
}
```

```
<p>Example of <em>plain text</em>
but with <em>even more fun mixed content</em></p>
```

```
{
   "p": {
      "em": [
         "plain text",
         "even more fun mixed content"
      ],
      "__text": "Example of \n but with"
   }
}
```

When Order Matters

Order matters ... especially when it *“doesn't matter”* (B. Tommie Usdin)

```
<p>It was getting hot. So hot, I could hardly stand it.</p>
<p>I woke up with a start. What a dream.</p>
```

```
...
```

When you don't know if it matters or not ...

... if sometimes it does, and sometimes is doesn't ...

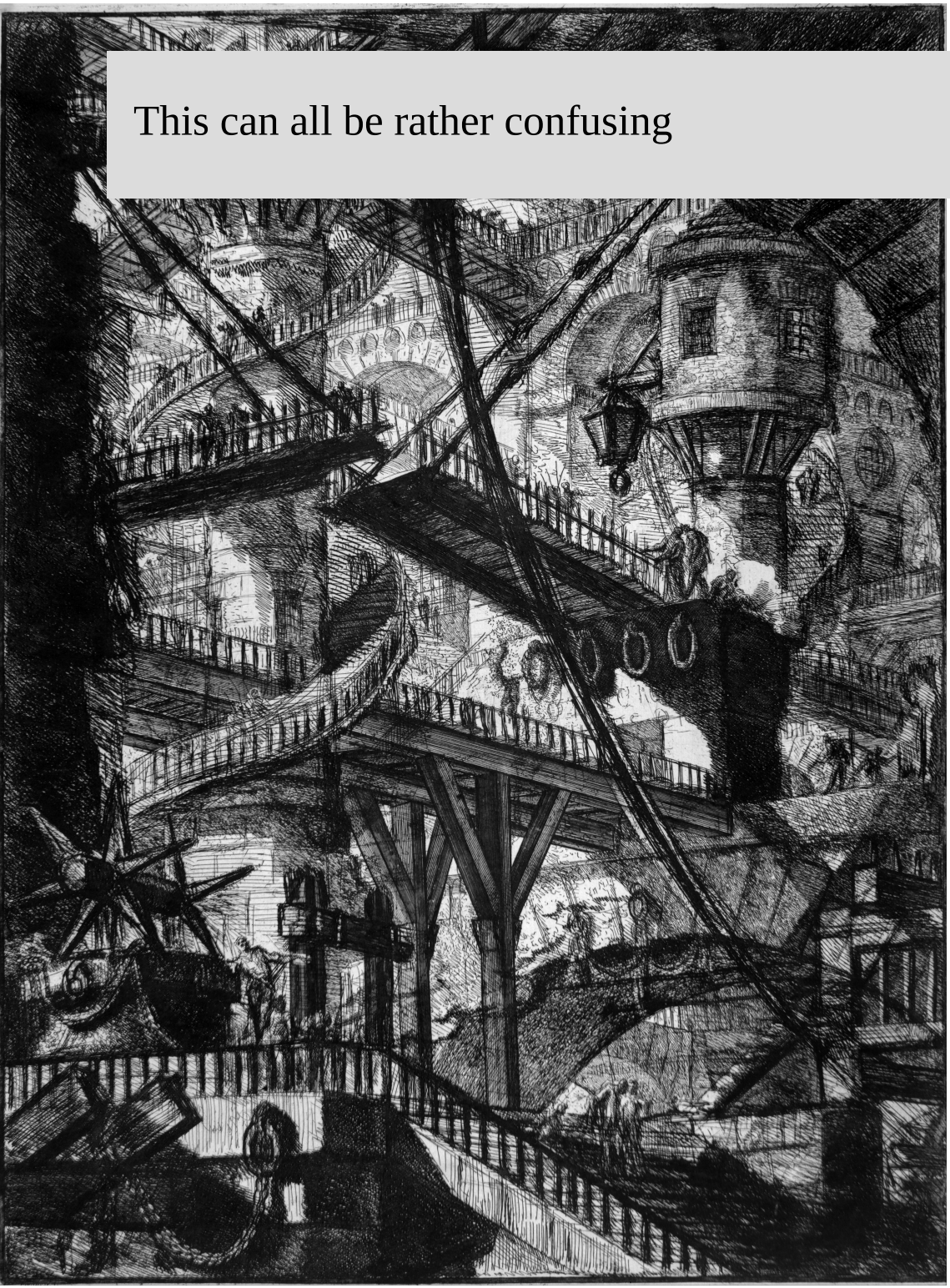
... isn't that as much as to say, it matters?

It's not supposed to matter -- kiss of death

```
<branch><leaf/></branch>
```

```
<branch><leaf/></branch>
```

This can all be rather confusing

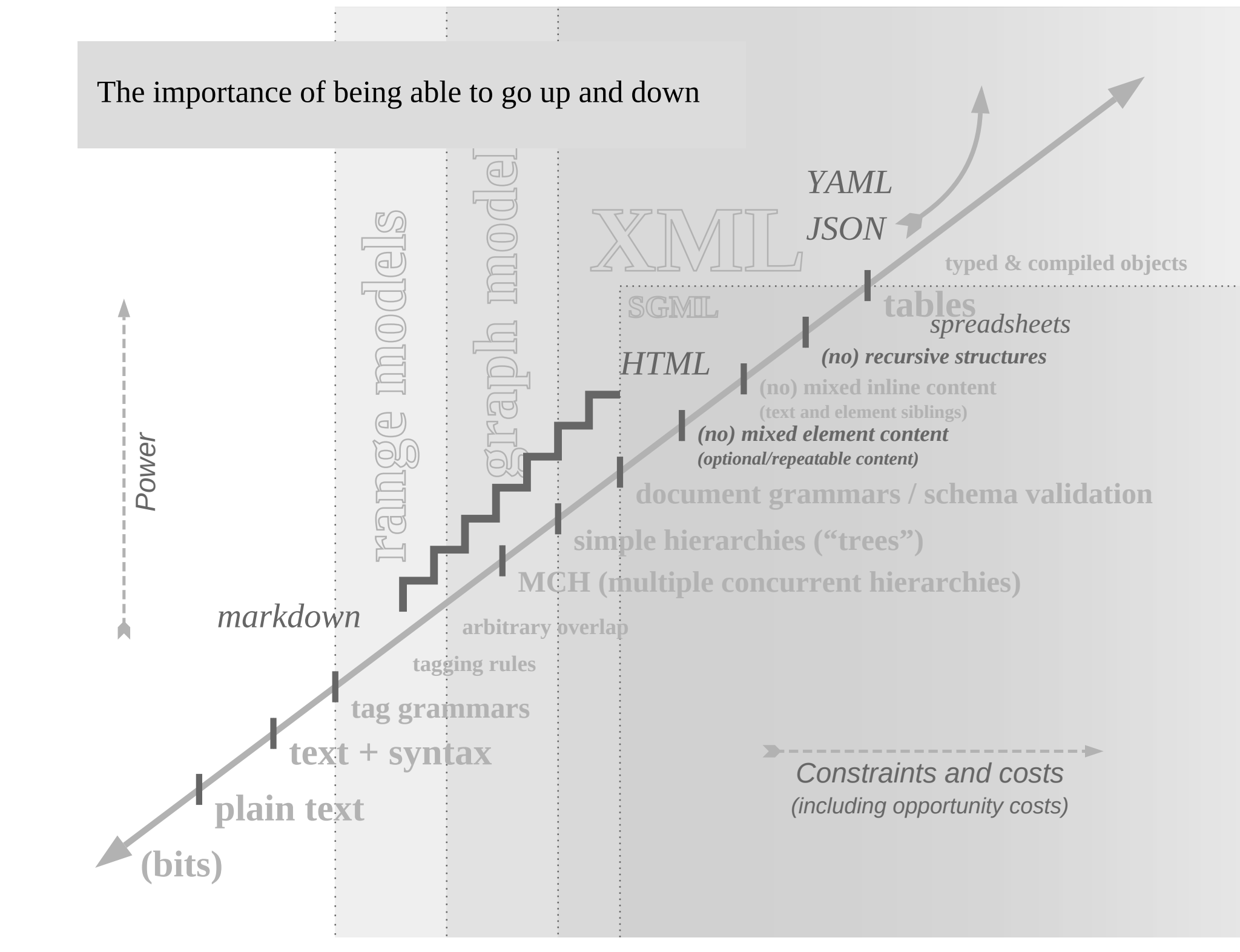


Robot Armies

We like XML because it's nutritious!

We like JSON because it's easy to digest!

We like YAML because there's no mess!


The importance of being able to go up and down
range models
graph model
XML
YAML
JSON
typed & compiled objects
SGML
tables
spreadsheets
HTML
(no) recursive structures
(no) mixed inline content
(text and element siblings)
(no) mixed element content
(optional/repeatable content)
document grammars / schema validation
simple hierarchies (“trees”)
MCH (multiple concurrent hierarchies)
arbitrary overlap
tagging rules
tag grammars
text + syntax
plain text
(bits)
markdown
Power
Constraints and costs
(including opportunity costs)

W

More / easier up and down the stairway

This means tools, parsers, utilities

Tolerance for *“foreign formats”*

More innovations to bridge the gap

How about *“dynamic markdown”*?

How about metaschemas?

How about Generic Spreadsheet Language?

Dynamic markdown

Map a conventional markdown syntax to a small abstract tag set

Add a convention for semantic tagging

Implicitly support structural induction (with maybe an escape hatch)

Then permit users/systems to declare their own semantic bindings

```
[*: em]
[**: b]
[()[]: a|@href]

## My note
```

```
.note Here's my note in my own personal markdown.
 It even has a (link)[http://example.com].

### Subsection

No markup, no semantics.

.special When we have markup ... we have semantics.

.quote "Those who have art and science, have
 religion. Those who have no art and no science --
 they can have religion." [Goethe]
```

Metaschemas

All mature tagging languages end up with metaschemas. Name your favorite tag set and there is likely to be some technology behind its schema maintenance.

How about metaschemas that are specifically built to bridge the gaps between formats?

Architectural Forms showed a way to do this with markup vocabularies

We can go even further if (for example) our metaschema enforces constraint sets for both XML and JSON, at the same time.

Generic Spreadsheet Language

There are millions of people who will never write ("plain-text based") code

But who *“program”* routinely in the form of spreadsheets

How about a generic spreadsheet language? We could call it GSML

It would be one step away from a User Interface

Could serve as a back end in / for (spreadsheet-ish) applications

And an interchange *“pivot”* format to richer semantics

As HTML is for web pages

Don't we already have this ... isn't it MS Excel?

Maybe we need an XSweet for Excel ... extract from Office Open .xslx to a clean intermediary ...

Wendell Piez

Balisage 2018

Rockville, Maryland