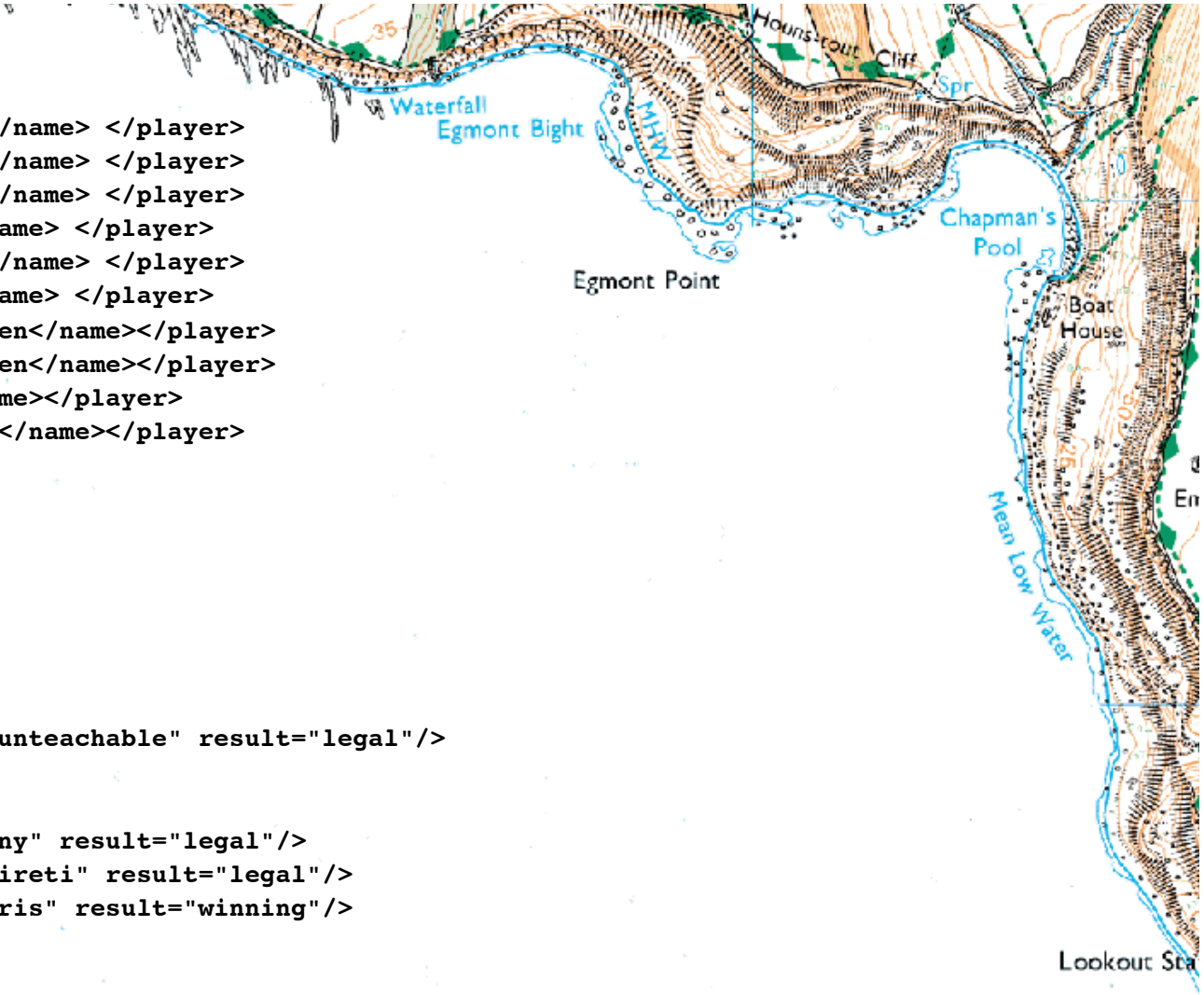
GML 4 requirements workshop Boulder TC September 2011

Carsten Rönsdorf

Head of Product and Data Engineering

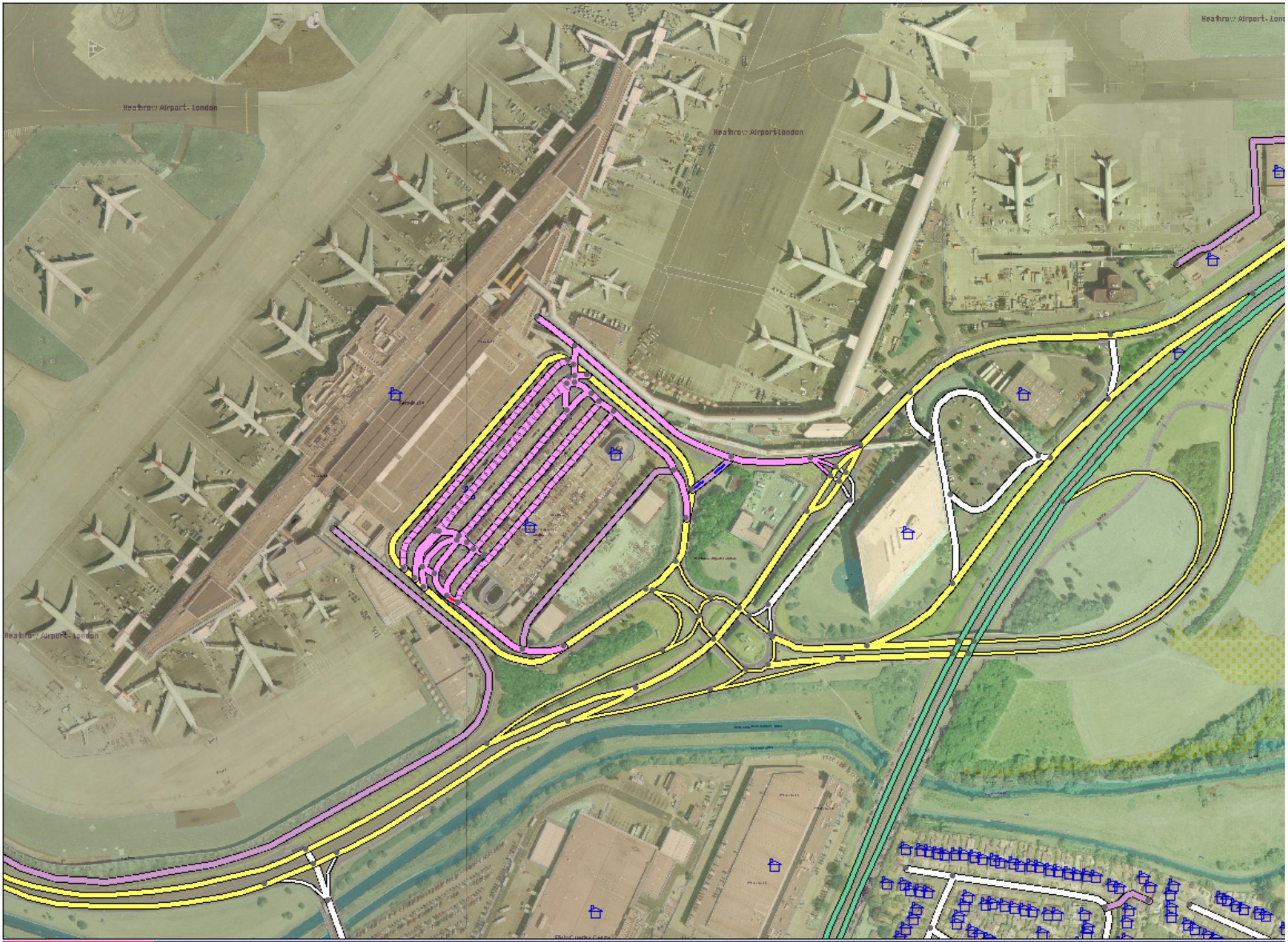
19 September 2011




```
<team id="wellFormed">
   <name>Well Formed</name>
      <players>
         <player id="jenny"> <name>Jenny</name> </player>
         <player id="ireti"> <name>Ireti</name> </player>
         <player id="chris"> <name>Chris</name> </player>
         <player id="mark"> <name>Mark</name> </player>
         <player id="peter"> <name>Peter</name> </player>
         <player id="phil"> <name>Phil</name> </player>
         <player id="stephen"><name>Stephen</name></player>
         <player id="carsten"><name>Carsten</name></player>
         <player id="matt"><name>Matt</name></player>
         <player id="debbie"><name>Debbie</name></player>
      </players>
</team>
<match>
   <teamA id="#wellFormed&Funded"/>
   <teamB id="#theLosers"/>
   <set winner="#wellFormed&Funded">
     <point winner="#wellFormed&Funded">
            <serve>
                        <server player="#unteachable" result="legal"/>
            </serve>
            <receive>
                        <dig player="#jenny" result="legal"/>
                        <volley player="#ireti" result="legal"/>
                        <spike player=#chris" result="winning"/>
            </receive>
     </point>
     <point winner="#wellFormed&Funded">
                        <serve>
                                    <server player="#mark" result="legal"/>
                        </serve>
                        <receive>
                                    <dig player="#rubbish" result="losing"/>

                        </receive>

     </point>
     <point winner="#wellFormed&Funded">
                        <serve>
```


Heathrow Airport - London
Heathrow Airport London

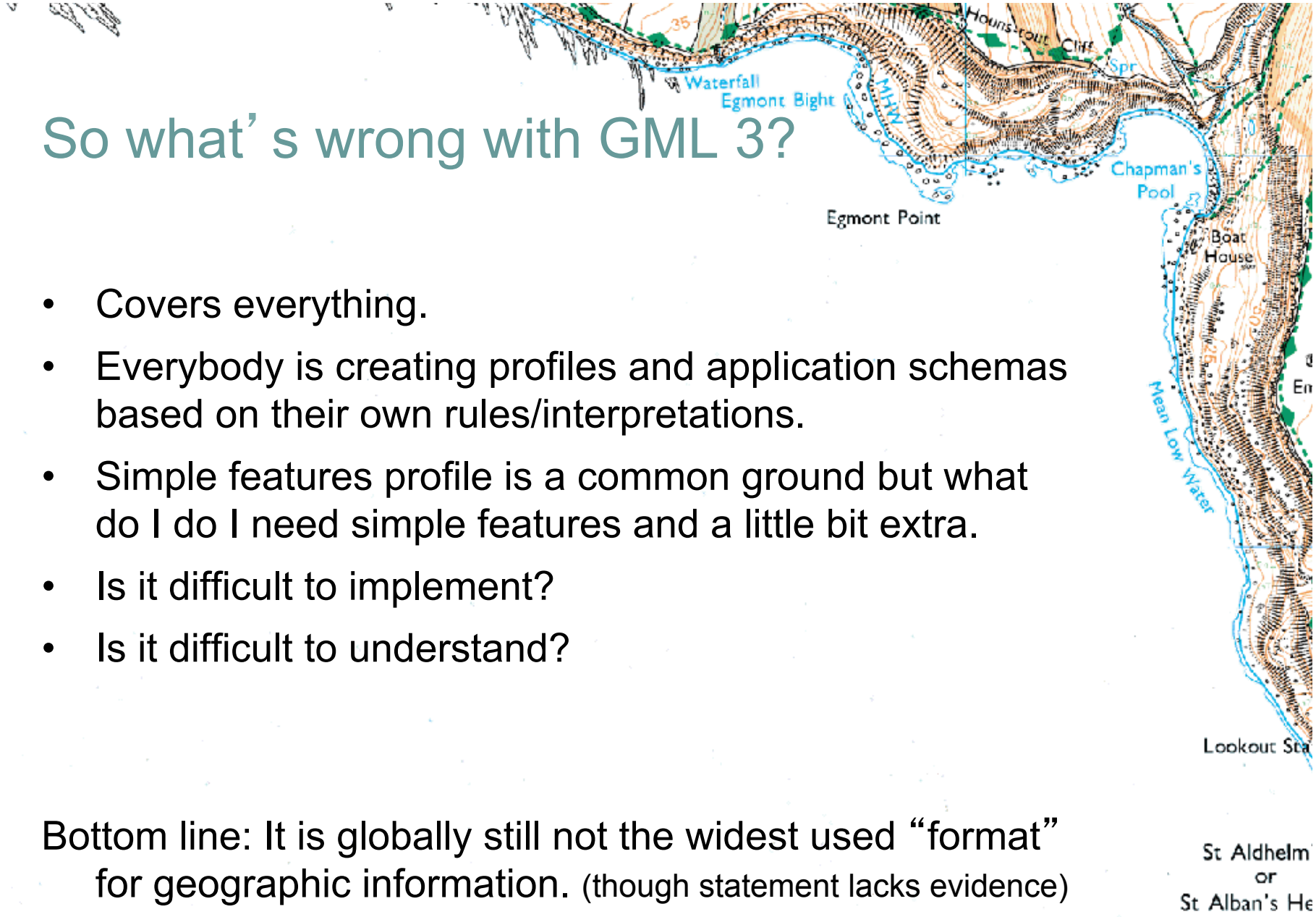
Success with GML?

- Flagship product OS MasterMap offered in GML only (November 2001).
- GML 2 that is.
- Translation software was not fit for purpose at that state.
- Commercial perspective: customer's want what's easiest for them and a big data management change was met with resistance.
- Long customer migration process until 2006/7: stick and carrot.
- Currently offer beta-version of one product in GML 3 to solicit market feedback.
- Big push towards GML 3 anticipated with Inspire.

Do users want to use GML?

YES NO

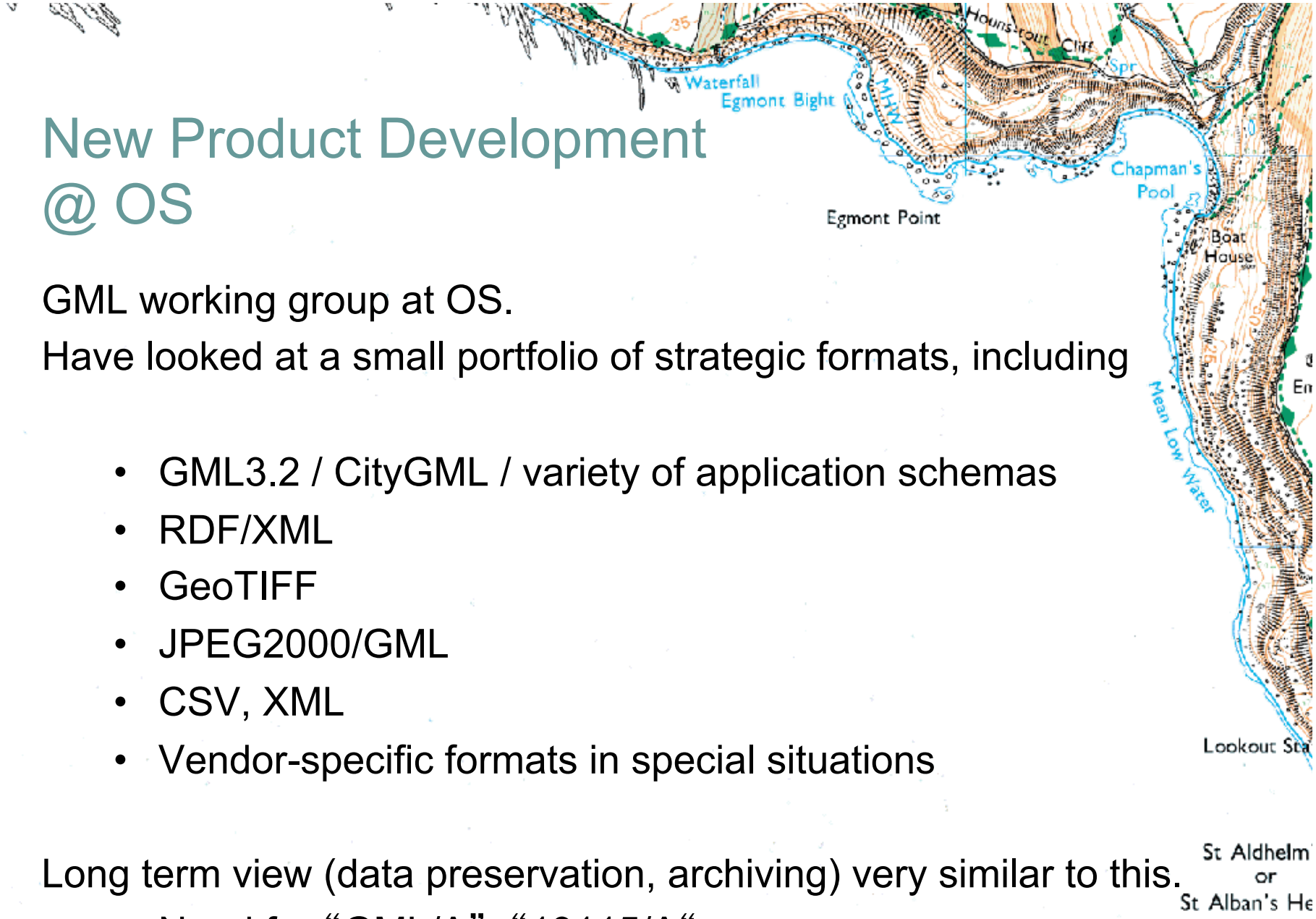
So what's wrong with GML 3?



- Covers everything.
- Everybody is creating profiles and application schemas based on their own rules/interpretations.
- Simple features profile is a common ground but what do I do I need simple features and a little bit extra.
- Is it difficult to implement?
- Is it difficult to understand?

Bottom line: It is globally still not the widest used "format" for geographic information. (though statement lacks evidence)

New Product Development @ OS



GML working group at OS.

Have looked at a small portfolio of strategic formats, including

- GML3.2 / CityGML / variety of application schemas
- RDF/XML
- GeoTIFF
- JPEG2000/GML
- CSV, XML
- Vendor-specific formats in special situations

Long term view (data preservation, archiving) very similar to this.

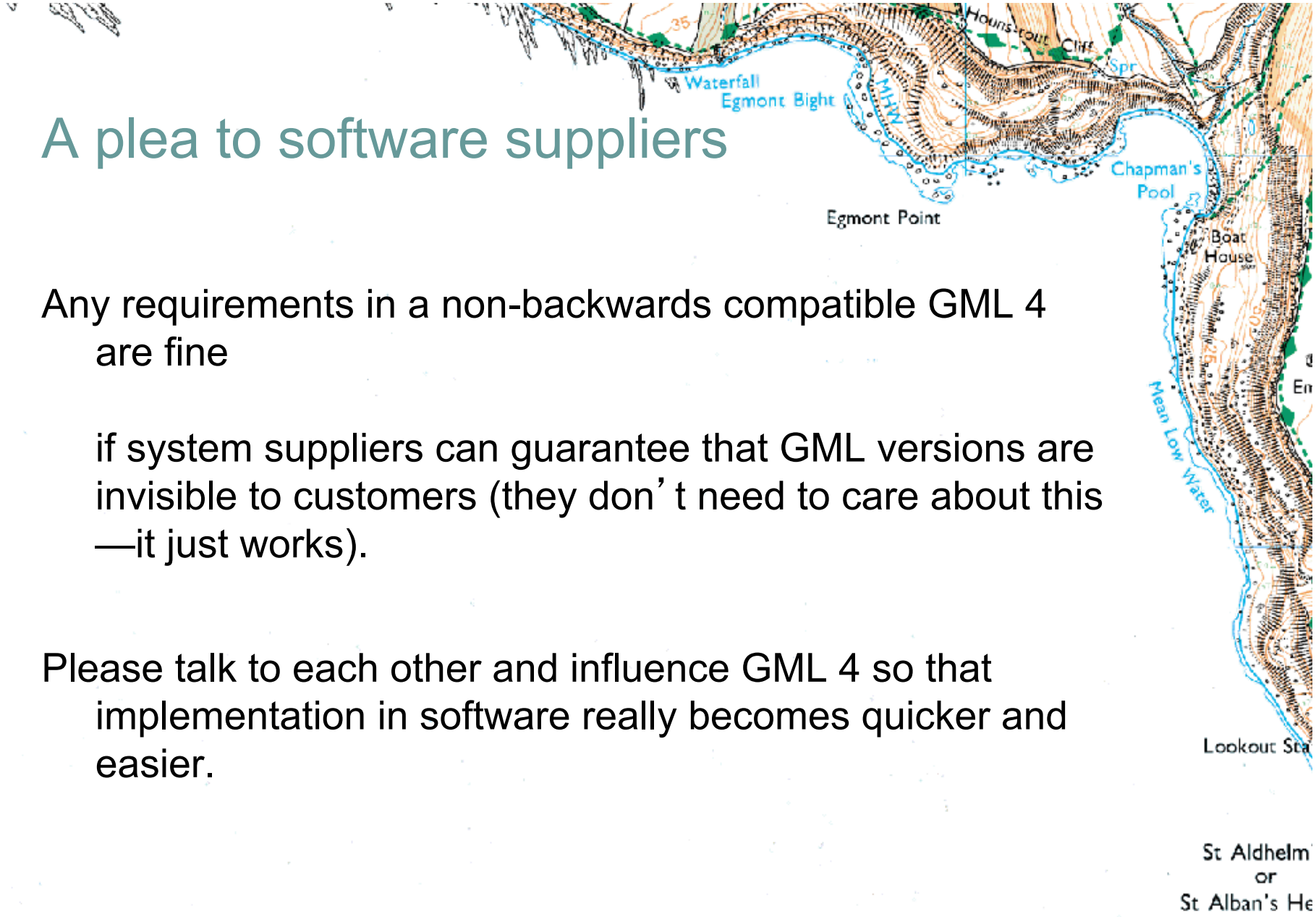
- Need for “GML/A”, “19115/A“
- and stronger integration of the two

If we were to introduce GML 4 (hypothetical)

- Assumption: GML 4 might be ready by December 2013
- Work with system suppliers throughout 2014 to agree support
- Publish first product by end of 2014 (But which ones? Maybe we should wait a few more years)
- Figure out if it makes sense/if we need to run GML3 and 4 in parallel. if not completely migrate customers for this first product by end of 2020
- Other products to follow suit

Verdict: Migration to a new data format/encoding may take a very long time.

A plea to software suppliers

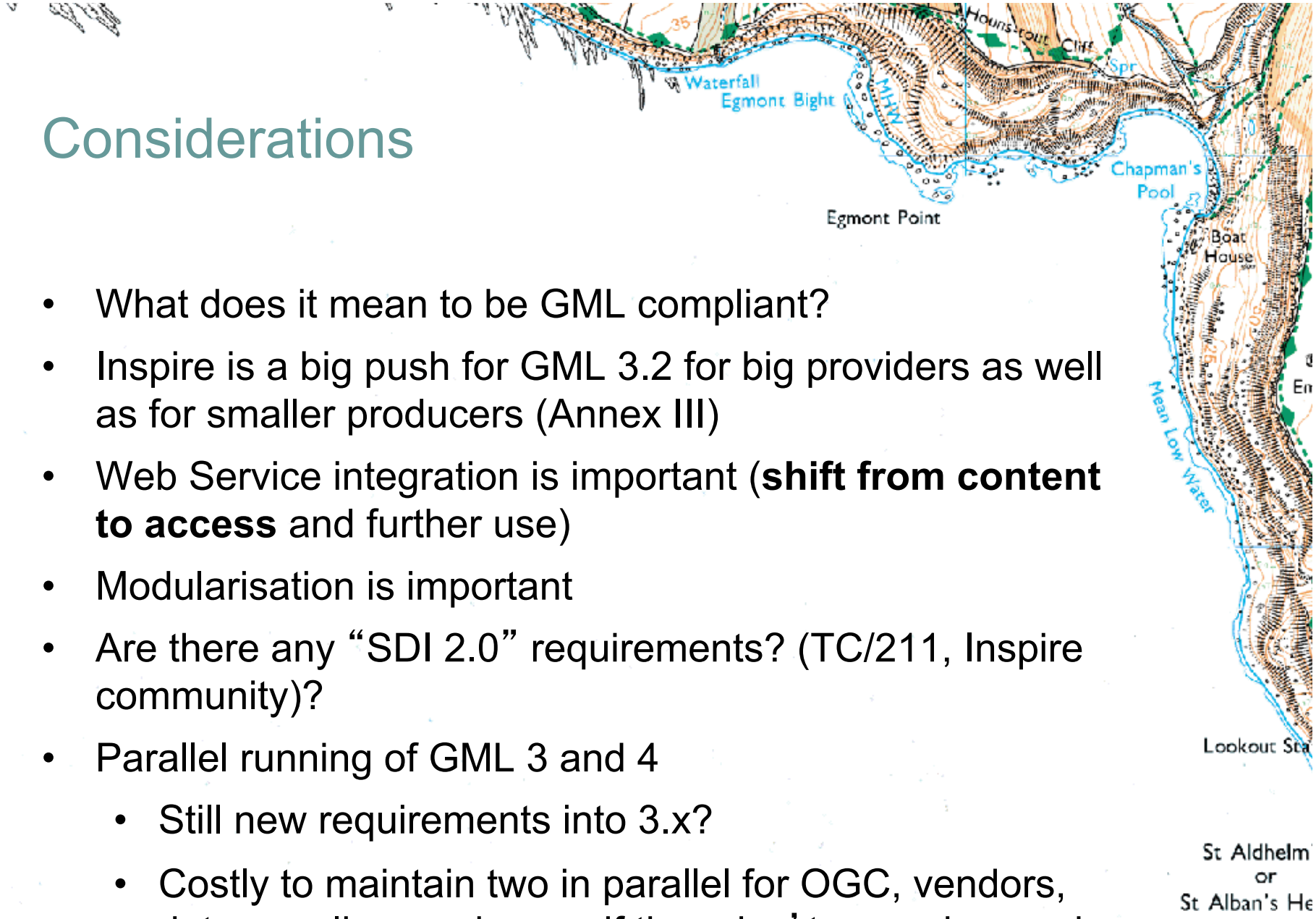


Any requirements in a non-backwards compatible GML 4 are fine

if system suppliers can guarantee that GML versions are invisible to customers (they don't need to care about this —it just works).

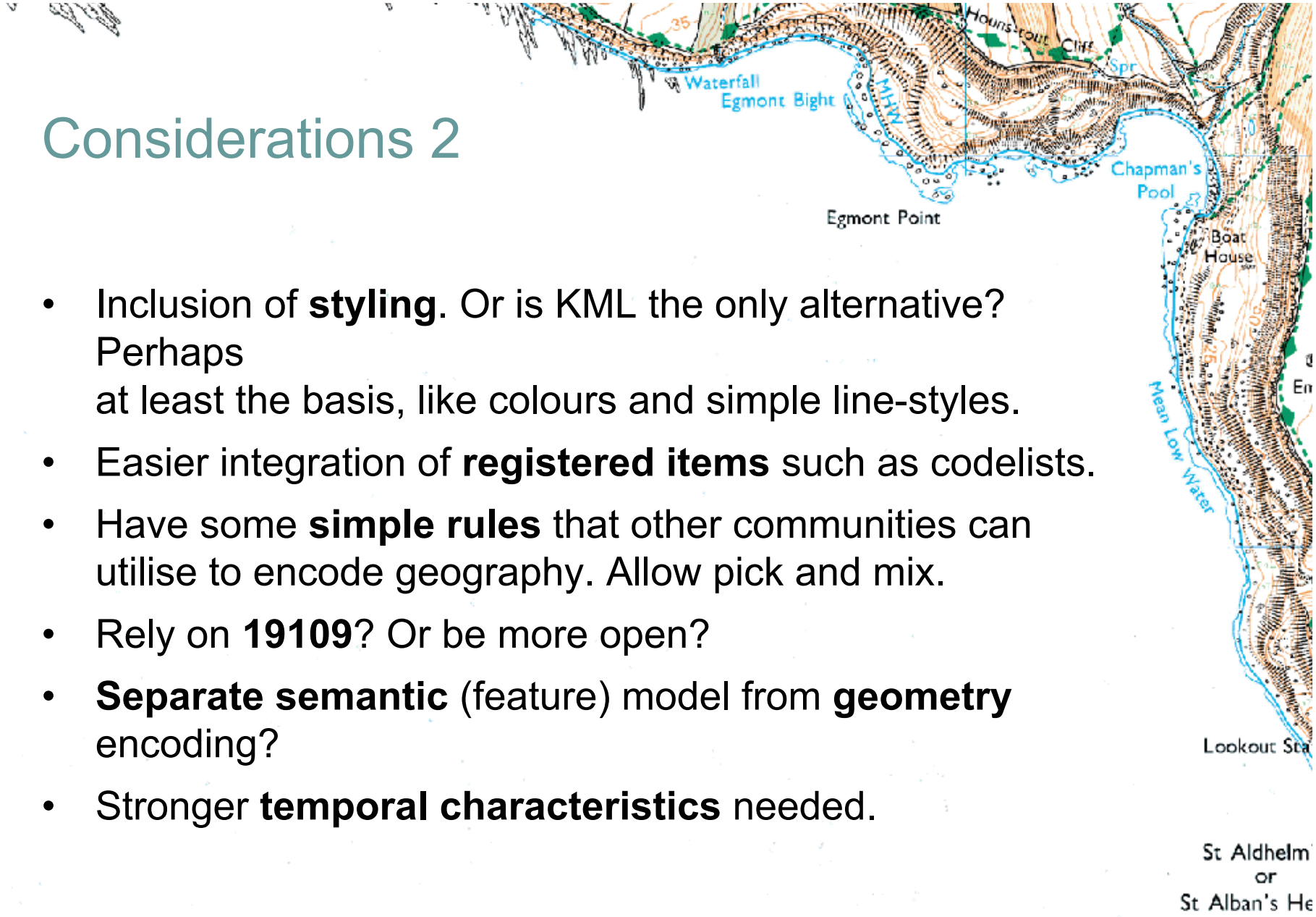
Please talk to each other and influence GML 4 so that implementation in software really becomes quicker and easier.

Considerations



- What does it mean to be GML compliant?
- Inspire is a big push for GML 3.2 for big providers as well as for smaller producers (Annex III)
- Web Service integration is important (**shift from content to access** and further use)
- Modularisation is important
- Are there any “SDI 2.0” requirements? (TC/211, Inspire community)?
- Parallel running of GML 3 and 4
  - Still new requirements into 3.x?
  - Costly to maintain two in parallel for OGC, vendors, data suppliers and users if there isn’t a good enough **differentiatior.**

Considerations 2

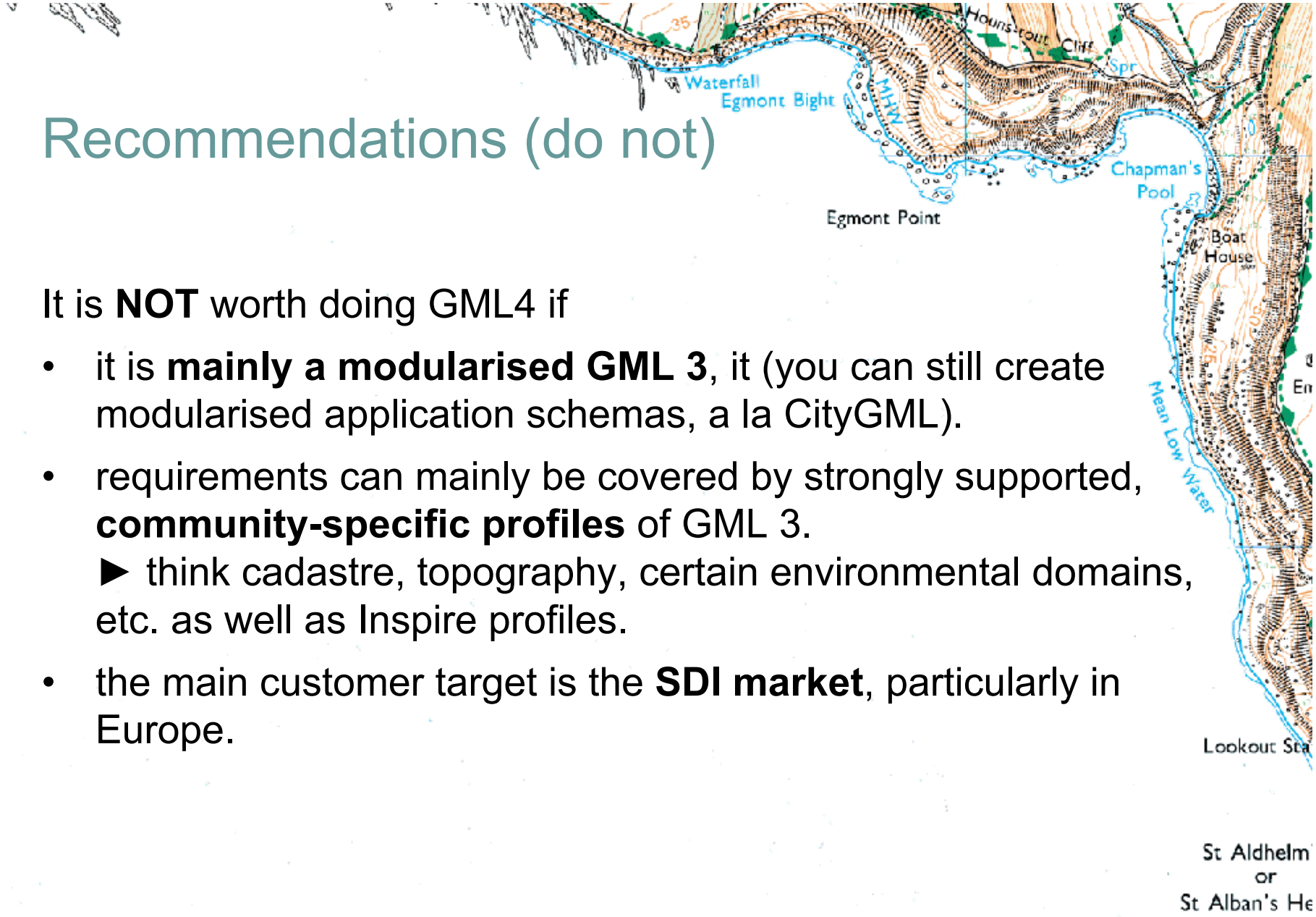


- Inclusion of **styling**. Or is KML the only alternative? Perhaps
  at least the basis, like colours and simple line-styles.
- Easier integration of **registered items** such as codelists.
- Have some **simple rules** that other communities can utilise to encode geography. Allow pick and mix.
- Rely on **19109**? Or be more open?
- **Separate semantic** (feature) model from **geometry** encoding?
- Stronger **temporal characteristics** needed.

KML, SKOS, RDF, JP2, ARML …

- Atom, JSON, RSS, …
- Ahhhhh.
- Modularise to make it **easy for other user communities to include bits** into their standards (we've already done this: GML can be put into JP2, RDF/XML, etc.)
- Push GML 4 as the **universal geographic encoding on the web**: true plug-and-play.
- GML is KML is RDF is ARML is ….
  Integrate by providing a **common, very open platform** that connects **effortlessly**.

Recommendations (do not)



It is **NOT** worth doing GML4 if

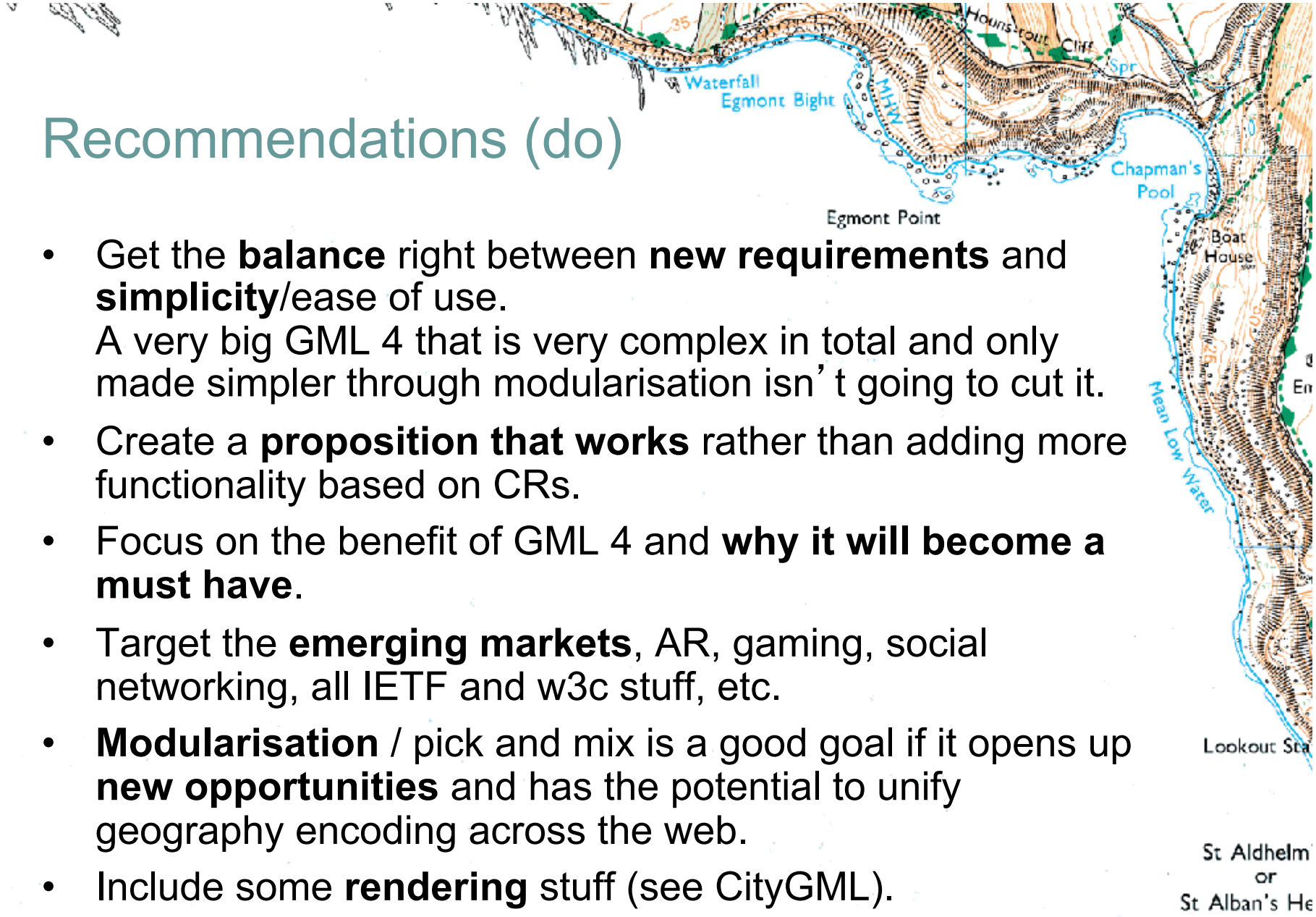
- it is **mainly a modularised GML 3**, it (you can still create modularised application schemas, a la CityGML).
- requirements can mainly be covered by strongly supported, **community-specific profiles** of GML 3.
  ► think cadastre, topography, certain environmental domains, etc. as well as Inspire profiles.
- the main customer target is the **SDI market**, particularly in Europe.

Differentiator?

What could it be in GML 4 that makes it worth having next to GML 3?

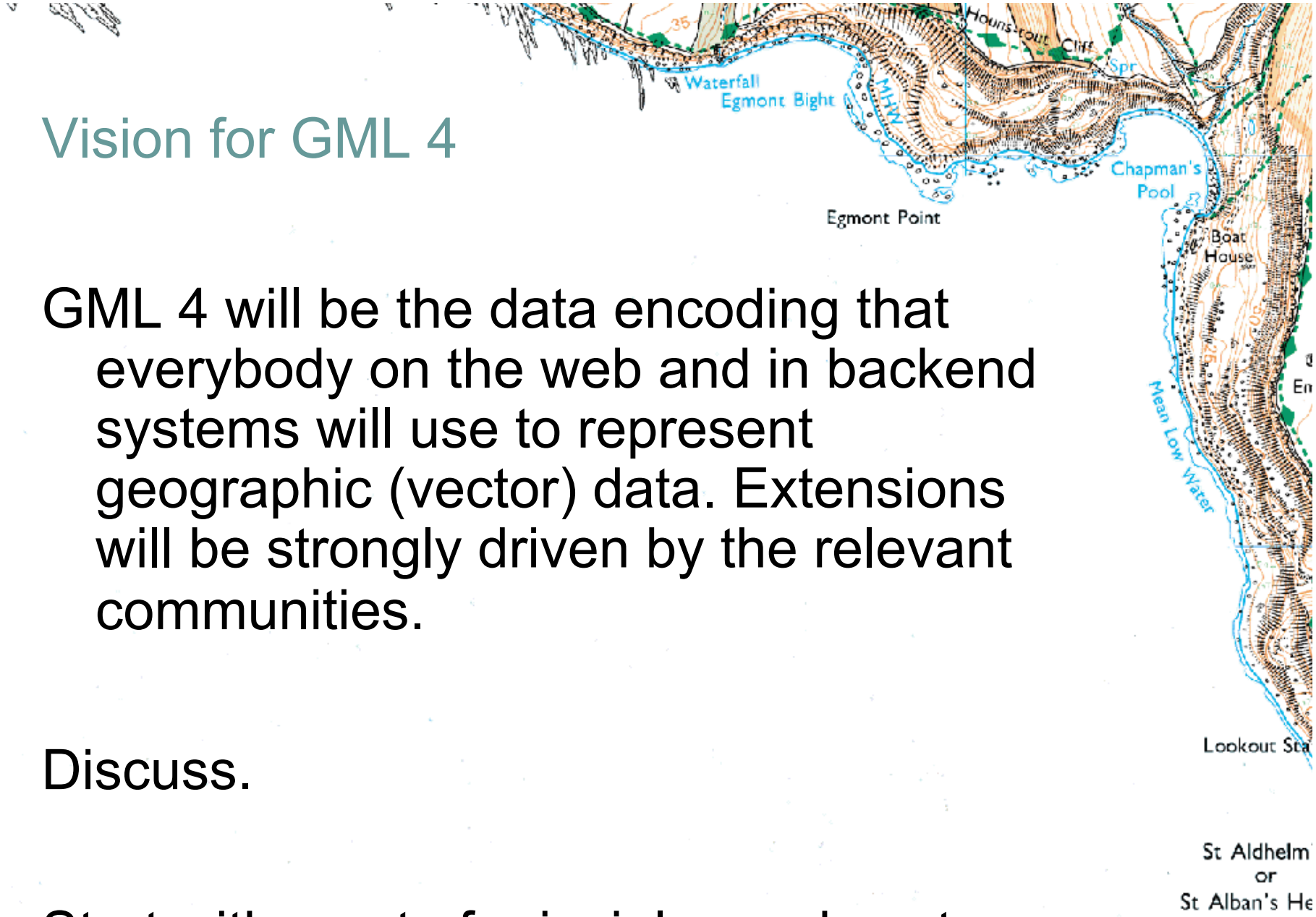
Or in fact SHP and others?

Recommendations (do)



- Get the **balance** right between **new requirements** and **simplicity**/ease of use.
  A very big GML 4 that is very complex in total and only made simpler through modularisation isn't going to cut it.
- Create a **proposition that works** rather than adding more functionality based on CRs.
- Focus on the benefit of GML 4 and **why it will become a must have**.
- Target the **emerging markets**, AR, gaming, social networking, all IETF and w3c stuff, etc.
- **Modularisation** / pick and mix is a good goal if it opens up **new opportunities** and has the potential to unify geography encoding across the web.
- Include some **rendering** stuff (see CityGML).
- Introduce **separate namespaces**.
- Integration of **coverages** could make sense.

Vision for GML 4



GML 4 will be the data encoding that everybody on the web and in backend systems will use to represent geographic (vector) data. Extensions will be strongly driven by the relevant communities.

Discuss.

Start with a set of principles on how to used GML 4.

Recommendation

Differentiate GML 4 from GML 3 and other common encodings and allow it to be easily utilised by other data communities on the web.