

Thoughts on Validating RDF Healthcare Data

David Booth, Ph.D.

KnowMED, Inc.

2013 W3C RDF Validation Workshop

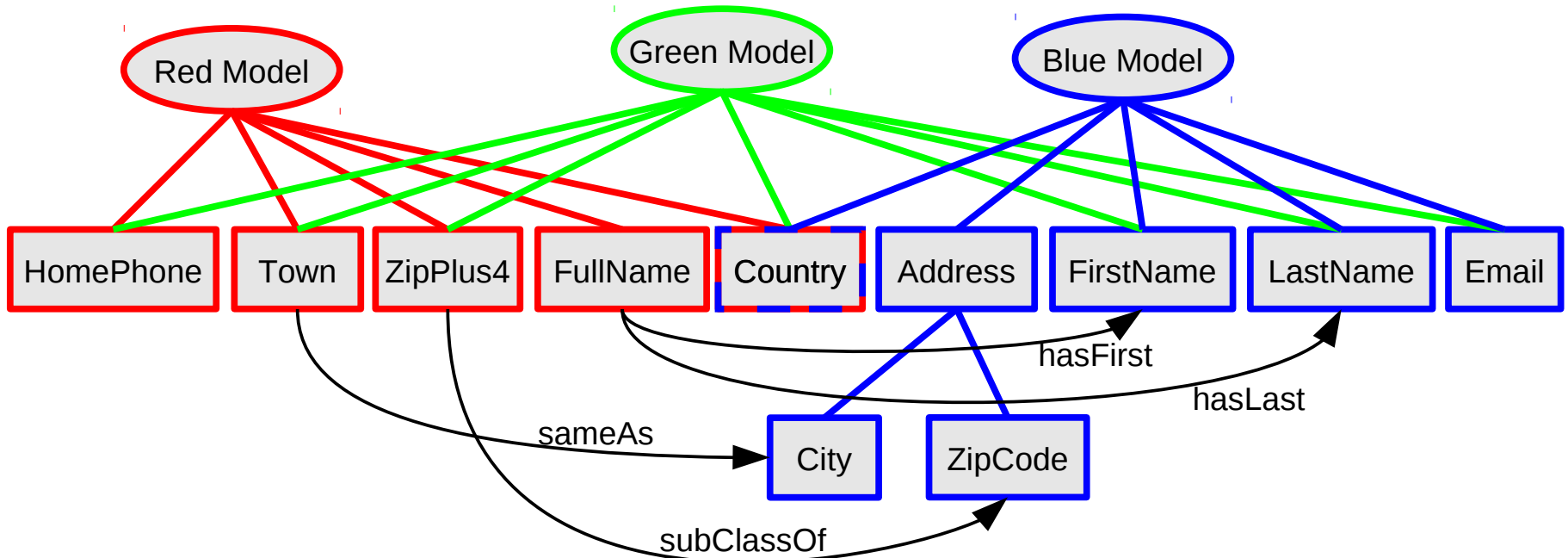
Latest version of these slides:

<http://dbooth.org/2013/validation/dbooth-slides.pdf>



Why RDF?

Schema promiscuous

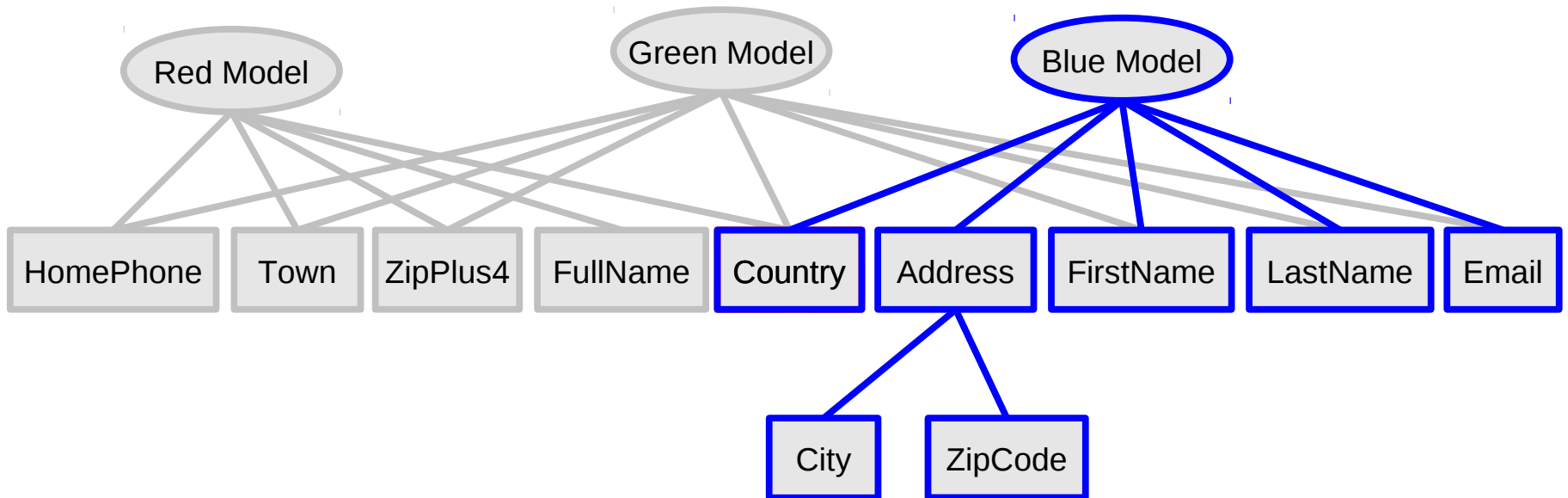


Multiple models peacefully coexist

Why RDF?

Schema promiscuous

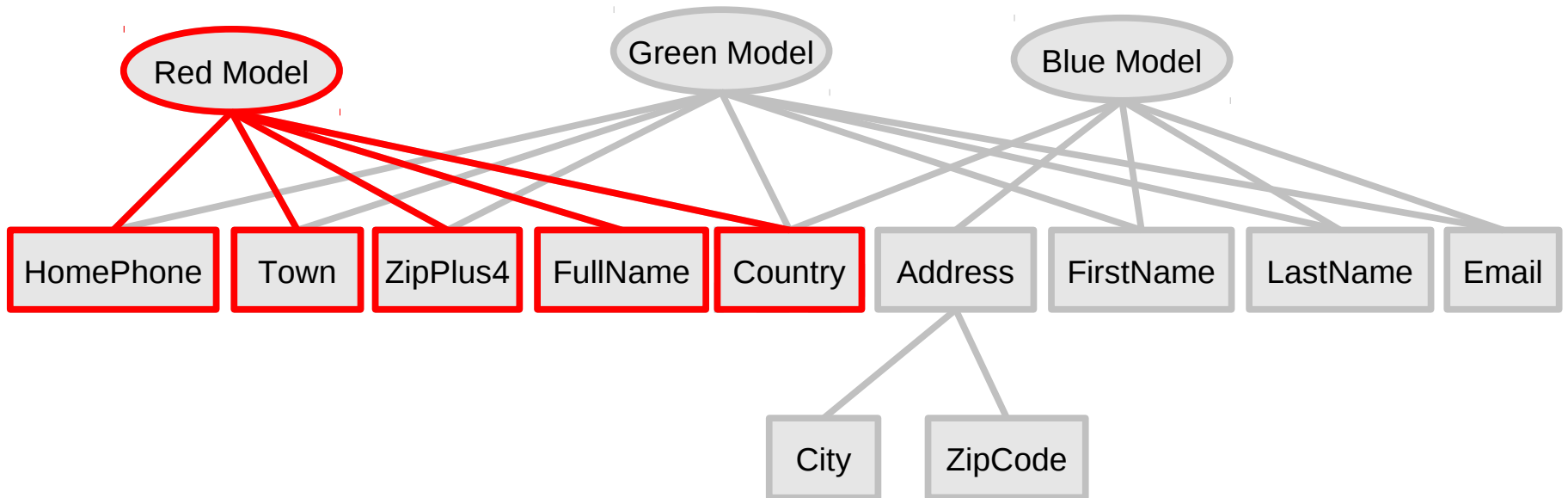
- What the Blue app sees:



Why RDF?

Schema promiscuous

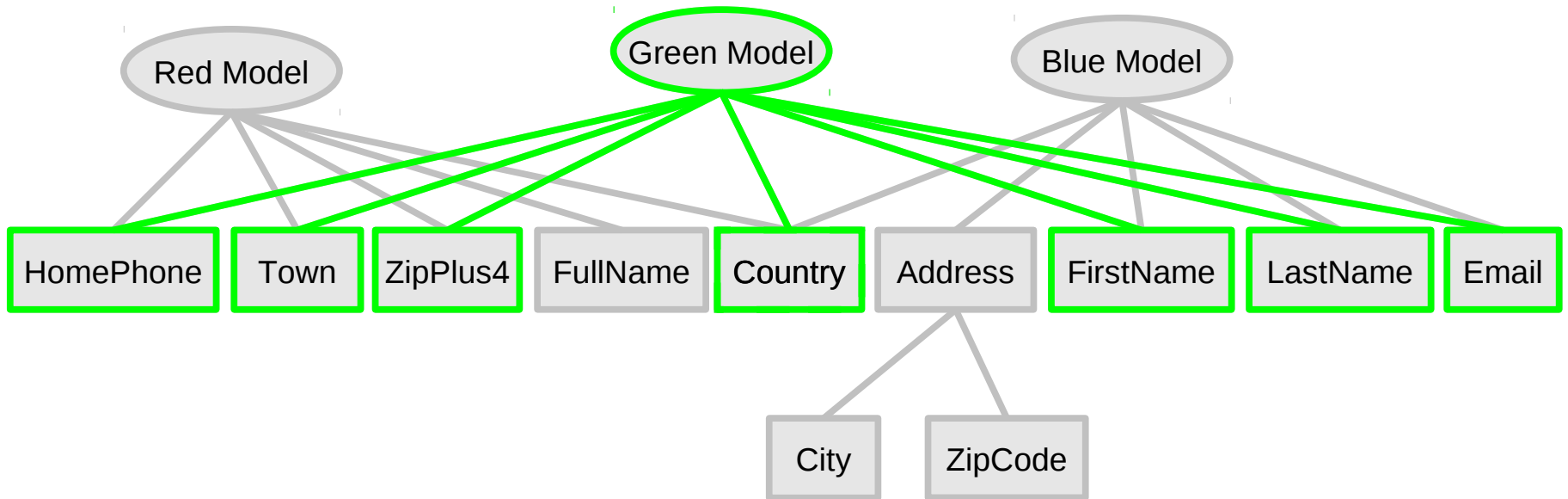
- What the Red app sees



Why RDF?

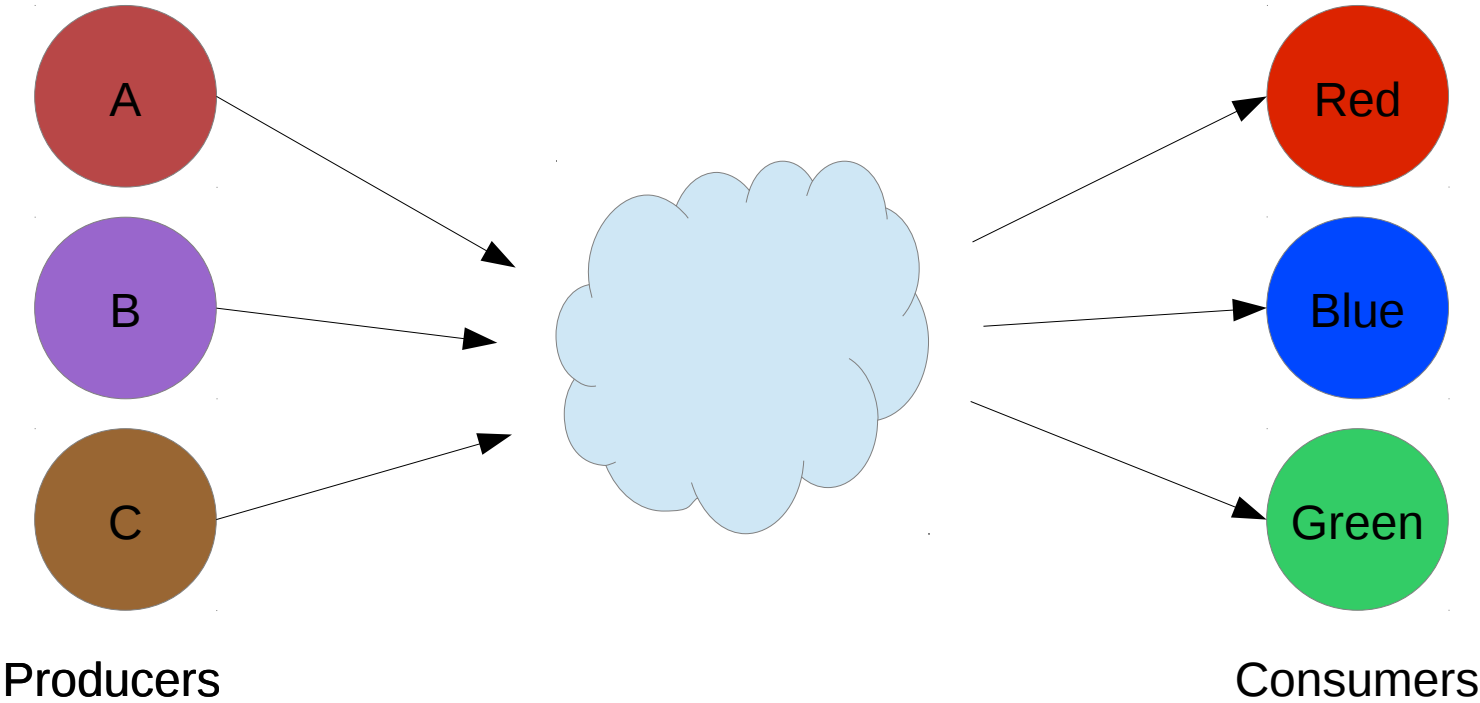
Schema promiscuous

- What the Green app sees



Need multiple validation perspectives on the same data!

Data producers and consumers



Two perspectives of validation

- Producers: **Model integrity**
 - Is the data well formed? (Sanity check)
 - Does it contain what I promised?
- Consumers: **Suitability for use**
 - Does the data meet my needs?
 - Different consumers have different needs!

Need multiple validation perspectives on the same data!

Features I'd like to see . . .

1. SPARQL-based framework

- Fewer languages == easier maintenance
- Nice to either:
 - Build on SPARQL, or
 - Use from SPARQL
- BUT if a new language were very concise and powerful, I'd jump on it.

2. Validation pipelines

- Simpler to write a series of SPARQL UPDATE operations than one big query
- Want standard ways to define validation pipelines

3. Better URI pattern matching and munging

- Often need to generate URIs from natural keys
- Want easier mechanisms for:
 - Checking URI patterns
 - Detecting misspellings

4. Validation like automated regression testing

- Lots of small, independent tests over one big one
 - E.g., one file per test
 - Contrast big ontology approach
- Goals:
 - Easy to add a new test
 - Can test anything

5. Operational versus declarative

- Declarative is convenient for very simple tests, e.g., pattern matching
- Operational is easier for more complex tests, e.g.:
 - "Do A, then B, then C, then result should be X"
- Note: SPARQL UPDATES can be used this way

Summary

- SPARQL-based
- Or something else that is powerful and concise