

# Open Source and Standards

Joe Gregorio  
Google

Zaheda Bhorat  
Google

# Scope

My opinions.

# A Poll

- Contribute to an open source project
- Contribute to a standard
- Both?

# Fair Warning

Turning the table

# Standards

*The nicest thing about standards is there are so many to choose from.*

- Andrew S. Tannenbaum

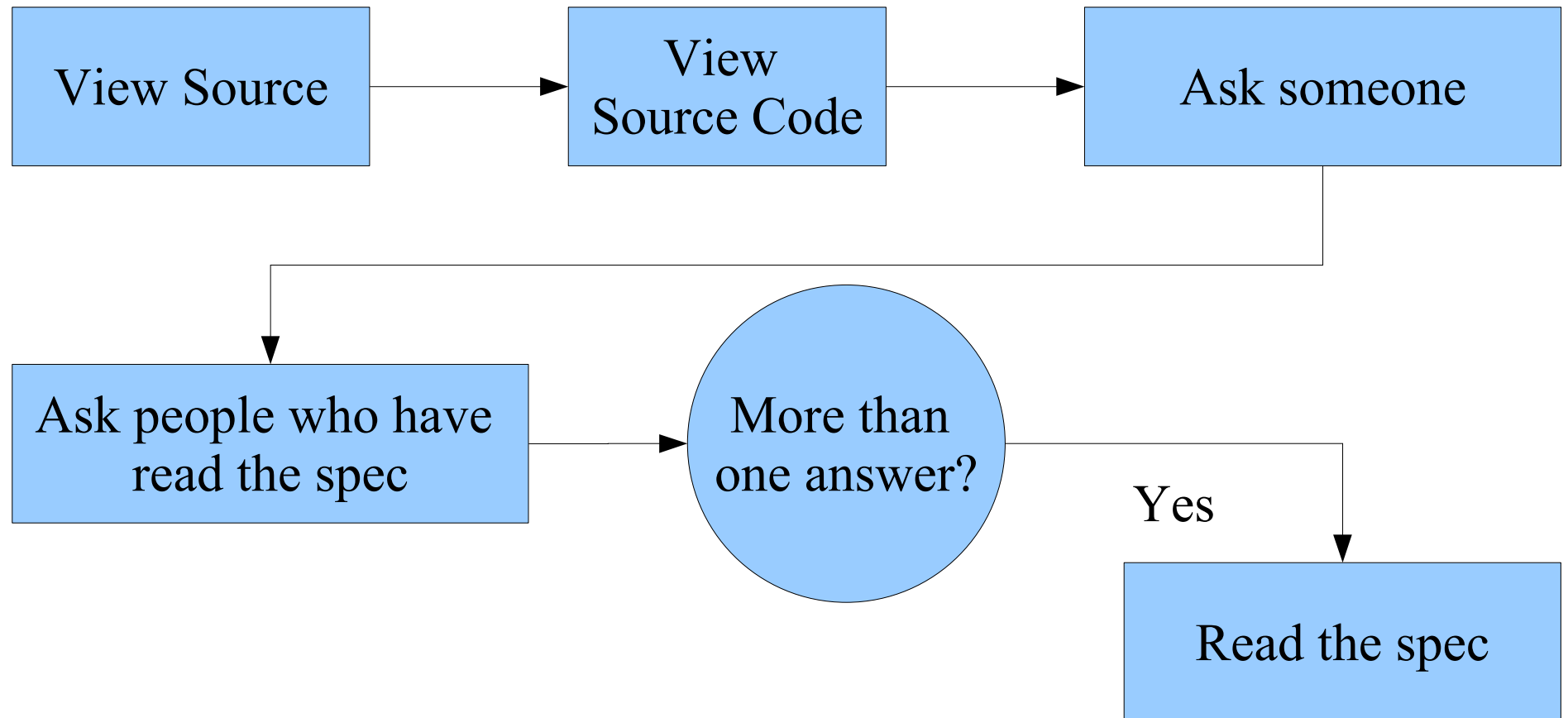
# Questions

- What are the ways Open Source and standards interact?
- What are the benefits of Open Source to standards
- What are the benefits of standards to Open Source?
- What can you as an Open Source developer do?

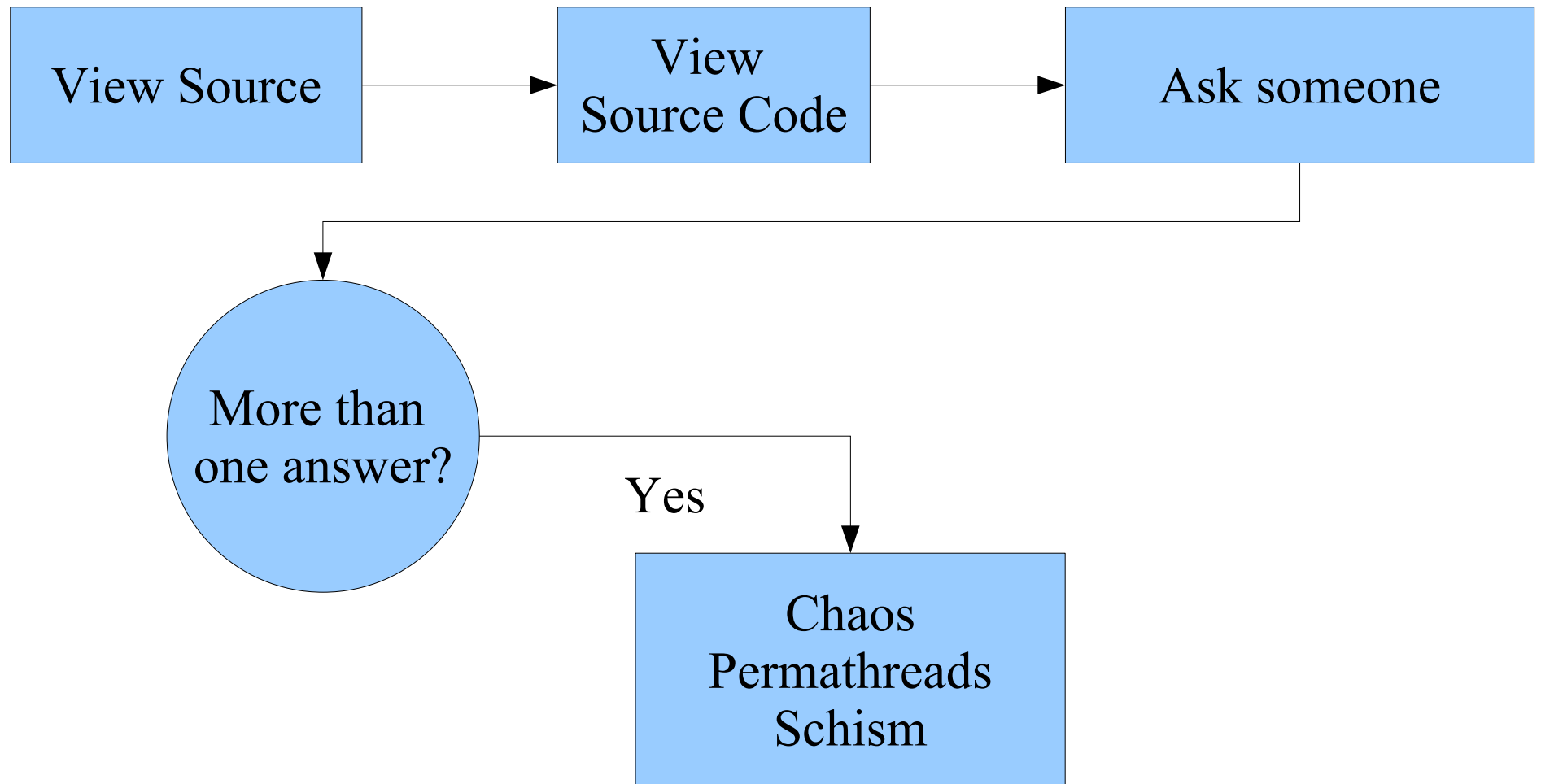
# Standards

What kinds of standards are we talking about?

# Implementation Question Resolution Heuristic



# Implementation Question Resolution Heuristic w/o Standards



# Example

*According to the RSS Advisory Board's Best Practices Profile, support for the enclosure element in RSS software varies significantly because of disagreement over whether the specification permits more than one enclosure per item. Although the author intended to permit no more than one enclosure in each item, this limit is not explicit in the specification.*

*For best support in the widest number of aggregators, an item should not contain more than one enclosure.*

<http://feedvalidator.org/docs/warning/DuplicateEnclosure.html>

# Kinds of spec text

1. Normative conformance requirement statements
2. Definitions
3. Informative statements and descriptions

# Specs vs. Code

Implementation – Code, View Source

Disagreements - Spec

# The kinds of code

Implementations

Validators/Test Suites

# Background

Atom – RFC 4287

AtomPub – RFC 5023

URI Templates

# Atom and AtomPub

Activity started on June 23, 2003

Started in IETF on June 16, 2004

Atom Syndication published Dec 2005

Atom Publishing Protocol published Oct 2007

Closed down on Oct 22, 2007

# More Background

## RSS

RDF Site Summary

Rich Site Summary

Really Simple Syndication

# IETF

Rough Consensus and Running Code

# Example

The RSS Validator

# Running Code

Cons: Maybe too much running code

# Running Code

*Amaya*

# Implementations

*Now, to (finally :P) answer your question. In the LJ Atom implementation, which was based on reading the Wiki and MT implementation and from conversations with the validator authors, I am using these mappings:*

- issued -> write time*
- created -> post time*
- modified -> modify time*

*- Evan Martin*

# Validators

## **Pros:**

Improve implementations

Focus discussion

## **Cons:**

Forks

# Lesson

Validator  $\neq$  Spec

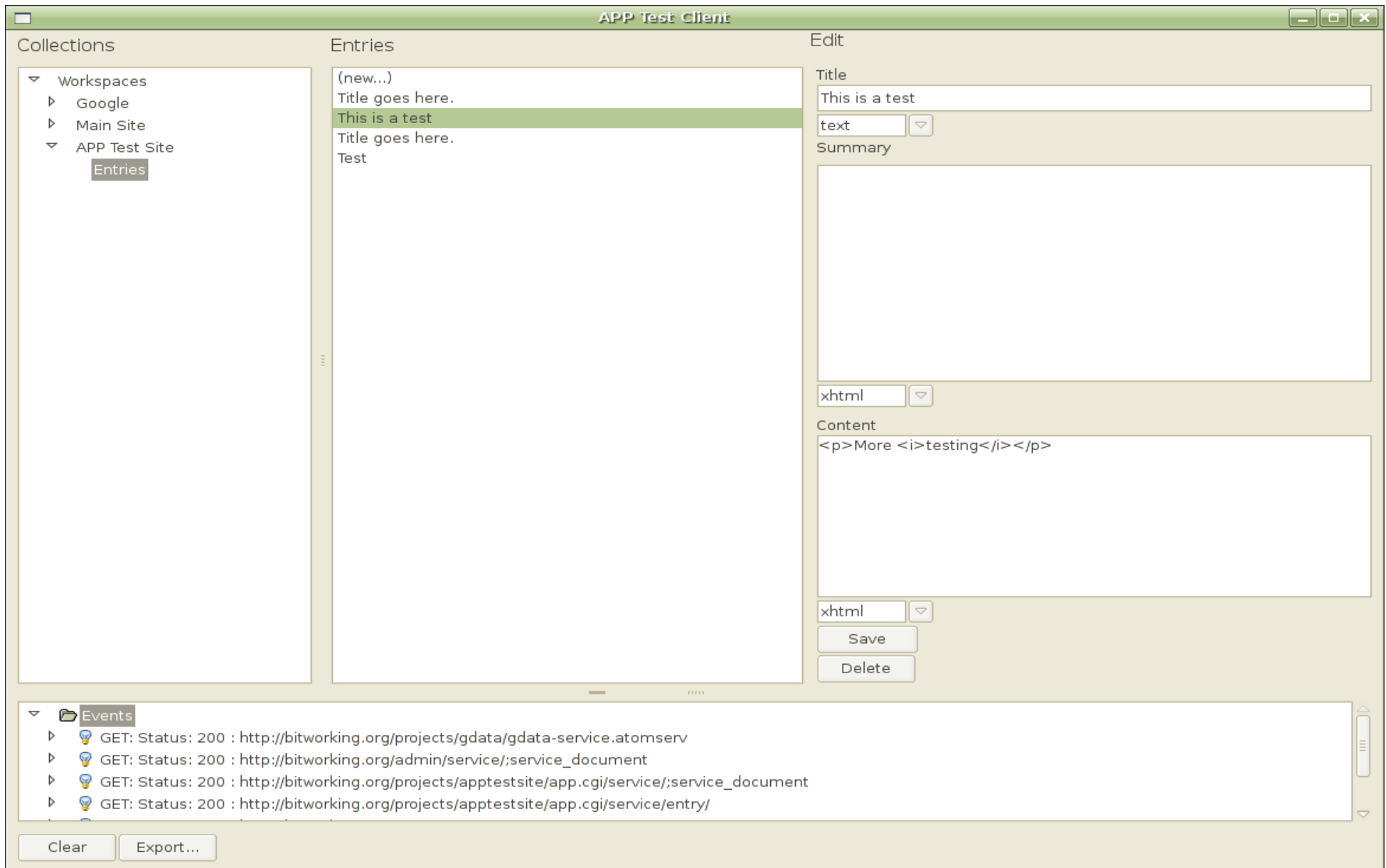
# Implementations

Porting implementations

# The Matrix

	A. Abdera (Snell)	B. Amplee (Peterson)	C. Blogger (Clinton)	D. AOL (Panzer)	E. LC Blogs (Snell)	F. LC Dogear (Snell)	G. Gregorio	H. Oracle (Mehta)	I. Six Apart (Reese)	J. O'Reilly (Greer)	K. WordPres: (Torres)	L. Roller (Johnson)	M. Propono (Johnson)	N. PHP+DB2 (Beyer)	O. Photo APP (Asakura)	P. O'Reilly Deli
1. APE (Bray)	loA1 (/)	loB1 X	loC1 X	loD1 (/)	loE1 (/)	loF1 (/)	loG1 (/)	loH1 (/)	loI1 -	loJ1 X	loK1 (/)	loL1 (/)	loM1 -	loN1 (/)	loO1 X	loP1 X
2. Abdera (Snell)	loA2 (/)	loB2 X	loC2 (/)	loD2 X	loE2 (/)	loF2 (/)	loG2 (/)	loH2 (/)	loI2 -	loJ2 (/)	loK2 (/)	loL2 (/)	loM2 -	loN2 (/)	loO2 X	loP2 (/)
3. O'Reilly (Fahlgren)	loA3 -	loB3 -	loC3 -	loD3 -	loE3 (/)	loF3 -	loG3 -	loH3 (/)	loI3 -	loJ3 (/)	loK3 -	loL3 -	loM3 -	loN3 (/)	loO3 -	loP3 -
4. Xameleon(Peterson)	loA4 -	loB4 -	loC4 -	loD4 -	loE4 -	loF4 -	loG4 -	loH4 -	loI4 -	loJ4 -	loK4 -	loL4 -	loM4 -	loN4 -	loO4 -	loP4 -
5. Gregorio	loA5 -	loB5 -	loC5 (/)	loD5 -	loE5 -	loF5 -	loG5 (/)	loH5 (/)	loI5 -	loJ5 -	loK5 (/)	loL5 -	loM5 -	loN5 (/)	loO5 -	loP5 -
6. Oracle (Mehta)	loA6 -	loB6 -	loC6 -	loD6 -	loE6 -	loF6 -	loG6 -	loH6 -	loI6 -	loJ6 -	loK6 -	loL6 -	loM6 -	loN6 -	loO6 -	loP6 -
7. Six Apart (Paschal)	loA7 -	loB7 -	loC7 X	loD7 -	loE7 -	loF7 -	loG7 X	loH7 -	loI7 -	loJ7 -	loK7 -	loL7 -	loM7 -	loN7 -	loO7 -	loP7 -
8. Windows Live Writer (Cheng)	loA8 -	loB8 -	loC8 (/)	loD8 -	loE8 (/)	loF8 X	loG8 (/)	loH8 (/)	loI8 -	loJ8 X	loK8 (/)	loL8 (/)	loM8 -	loN8 (/)	loO8 -	loP8 -
9. Flock (Loisant)	loA9 -	loB9 -	loC9 -	loD9 -	loE9 -	loF9 -	loG9 -	loH9 -	loI9 -	loJ9 -	loK9 -	loL9 -	loM9 -	loN9 -	loO9 -	loP9 -
10. Propono (Johnson)	loA10 -	loB10 -	loC10 (/)	loD10 -	loE10 (/)	loF10 -	loG10 X	loH10 X	loI10 -	loJ10 -	loK10 (/)	loL10 (/)	loM10 (/)	loN10	loO10 -	loP10 -
11. Nukular (McCulley)	loA11 -	loB11 (/)	loC11 (/)	loD11 X	loE11 -	loF11 -	loG11 (/)	loH11 (/)	loI11 -	loJ11 -	loK11 (/)	loL11 (/)	loM11 -	loN11 (/)	loO11 (/)	loP11 -
12. Javascript (Beyer)	loA12 -	loB12 -	loC12 -	loD12 -	loE12 -	loF12 -	loG12 -	loH12 -	loI12 -	loJ12 -	loK12 -	loL12 -	loM12 -	loN12 -	loO12 -	loP12 -
13. C# (Asakura)	loA13 -	loB13 -	loC13 -	loD13 -	loE13 -	loF13 -	loG13 X	loH13 -	loI13 -	loJ13 X	loK13 -	loL13 -	loM13 -	loN13 -	loO13 -	loP13 (/)
14. WSO2 (Marsh)	loA14 -	loB14 -	loC14 (/)	loD14 -	loE14 -	loF14 -	loG14 -	loH14 -	loI14 -	loJ14 -	loK14 -	loL14 -	loM14 -	loN14 -	loO14 -	loP14 -
15. Google data py	loA15 -	loB15 -	loC15 (/)	loD15 -	loE15 -	loF15 -	loG15 (/)	loH15 -	loI15 -	loJ15 -	loK15 -	loL15 -	loM15 -	loN15 (/)	loO15 -	loP15 -

# wxPython



# URI Templates

```
http://example.org/{userid}
```

```
userid = 'fred'
```

```
http://example.org/fred
```

# Why it matters

Why you should get involved?

Better specs

Better software

Less time arguing

# Open Source and Standards

Joe Gregorio  
Google

Zaheda Borat  
Google

# Scope

My opinions.

# A Poll

- Contribute to an open source project
- Contribute to a standard
- Both?

# Fair Warning

Turning the table

# Standards

*The nicest thing about standards is there are so many to choose from.*

- Andrew S. Tannenbaum

5

You may not have high opinions of standards, or the standards process.

You are swimming in standards. From your wi-fi, ethernet, tcp/ip, http, html, possibly your programming language, file formats (Atom, ODF, etc)

As an Open Source developer it is important for **interop**.

## Questions

- What are the ways Open Source and standards interact?
- What are the benefits of Open Source to standards
- What are the benefits of standards to Open Source?
- What can you as an Open Source developer do?

6

The goal of this talk is to answer these questions

(and then towards the end find out how this does or doesn't work).

# Standards

What kinds of standards are we talking about?

7

Not - Ad hoc

Institutional control

- stability – IETF – Once an RFC is published that document will stay there, unchanged, forever. Newer specs may come along and supplant or obsolete that spec, but when I refer to the old spec (for example: RFC 822) it will always mean the same thing.

- Errata process, decision making process, conflict resolution, etc.

Good specs will have three kinds of text:

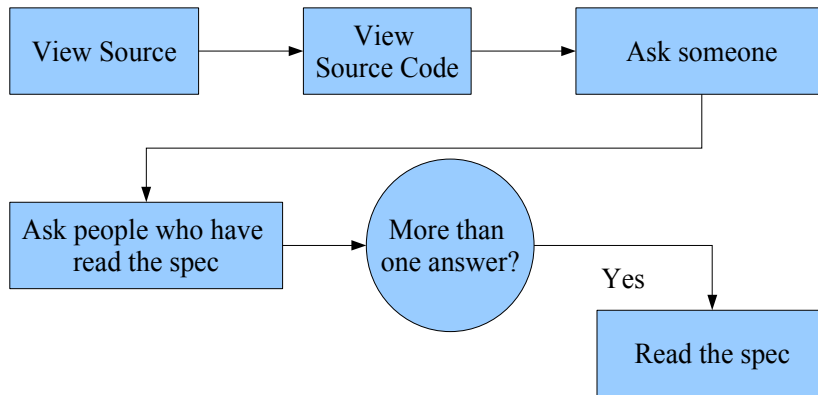
- normative text

- examples

- non-normative (chatty) text that gives a context and rationalization for the decisions encoded in the normative text, and also implementation hints.

- Specs are targeted at the implementer, not the consumer. Reference guides, busy developer guides, etc will always be needed and are an important part the eco-system.

## Implementation Question Resolution Heuristic



8

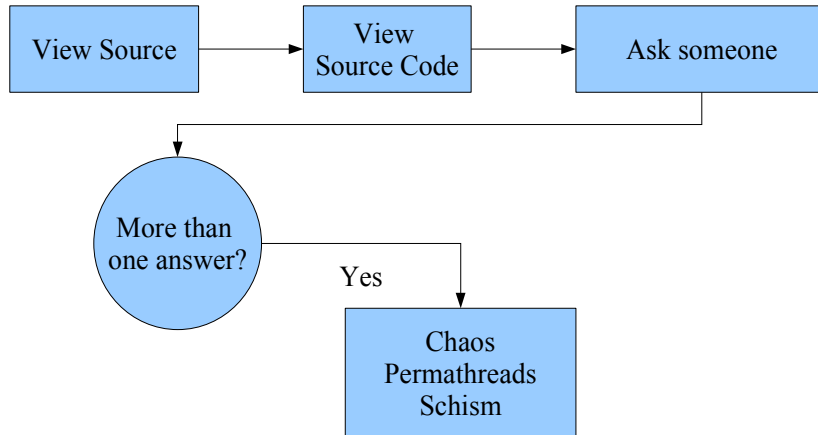
Note that the only time you go back to the spec is to resolve a disagreement.

Open source implementations play an important part in this process.

Will look at source code to an open source project, presuming you work for a company where the GPL won't cause your face to melt off.

Note that if you still get different answers after reading the spec then that may feed into the errata process.

# Implementation Question Resolution Heuristic w/o Standards



## Example

*According to the RSS Advisory Board's Best Practices Profile, support for the enclosure element in RSS software varies significantly because of disagreement over whether the specification permits more than one enclosure per item. Although the author intended to permit no more than one enclosure in each item, this limit is not explicit in the specification.*

*For best support in the widest number of aggregators, an item should not contain more than one enclosure.*

<http://feedvalidator.org/docs/warning/DuplicateEnclosure.html>

10

The importance of specs

## **Kinds of spec text**

1. Normative conformance requirement statements
2. Definitions
3. Informative statements and descriptions

## Specs vs. Code

Implementation – Code, View Source  
Disagreements - Spec

12

Show of hands, how many people have read RFC 2616 word for word?

The role of specifications isn't what you think.

Code is almost never developed 'against' a specification.

It is written and tested against an impl, and the specification is used for 'resolving disagreements'.

This is true not just in software, I used to write firmware for RFID readers, and to read a new tag you'd get a copy of said tag, setup the initial waveform and ping the tag and look at the response.

So all the “issues” you've seen with standardization, they're not just in software.

# The kinds of code

Implementations  
Validators/Test Suites

13

Let's talk about the different kinds of software projects you may see.

Implementations (client, server, reference, experimental)

Validators like the 'feed validator'

# Background

Atom – RFC 4287  
AtomPub – RFC 5023  
URI Templates

14

Atom and AtomPub came from a contentious background, and this may have influenced the standards process. It is from this contentious background that the atom WG was formed

URI Templates (OpenSearch, WADL)

## **Atom and AtomPub**

Activity started on June 23, 2003

Started in IETF on June 16, 2004

Atom Syndication published Dec 2005

Atom Publishing Protocol published Oct 2007

Closed down on Oct 22, 2007

15

(for AtomPub) Syndication format was shorter  
they couldn't be a parallel process

# More Background

## RSS

**RDF Site Summary**  
**Rich Site Summary**  
**Really Simple Syndication**

16

RSS – not really ad hoc, but definitely not an institutional standard.

9 incompatible versions of RSS, example of an anti-standard

# IETF

## Rough Consensus and Running Code

17

IETF rules try to help define 'rough consensus.'

Considered critical to interop to have working implementations.

## Example

The RSS Validator

18

Our first example of open source and standards, even before we get to Atom.

Way to focus conversation around ad-hoc standards.

# Running Code

Cons: Maybe too much running code

19

The group was under a misapprehension about what counted as 'running code', and so Blogger implemented Atom 0.3, we went from a couple blogs to millions.

That was a little too much.

Led some developers to believe they were **done** supporting Atom.

The 0.3 version lasted until around August of 2006 (RFC 4287 was published in Dec 2005).

# Running Code

Amaya

20

Too little. Wrong time.

# Implementations

*Now, to (finally :P) answer your question. In the LJ Atom implementation, which was based on reading the Wiki and MT implementation and from conversations with the validator authors, I am using these mappings:*

- issued -> write time*
- created -> post time*
- modified -> modify time*

*- Evan Martin*

21

Here the implementation is based on view source of another open source implementation.

And that porting work is producing feedback into the spec.

# Validators

**Pros:**

- Improve implementations

- Focus discussion

**Cons:**

- Forks

22

The problem with open source validators is of forks, tracking non-quite-specifications, is a fork.

and a fork for political reasons

# Lesson

Validator != Spec

# Implementations

Porting implementations

24

Once you're past the initial stages of experimenting open source implementations help with spreading the technology around.

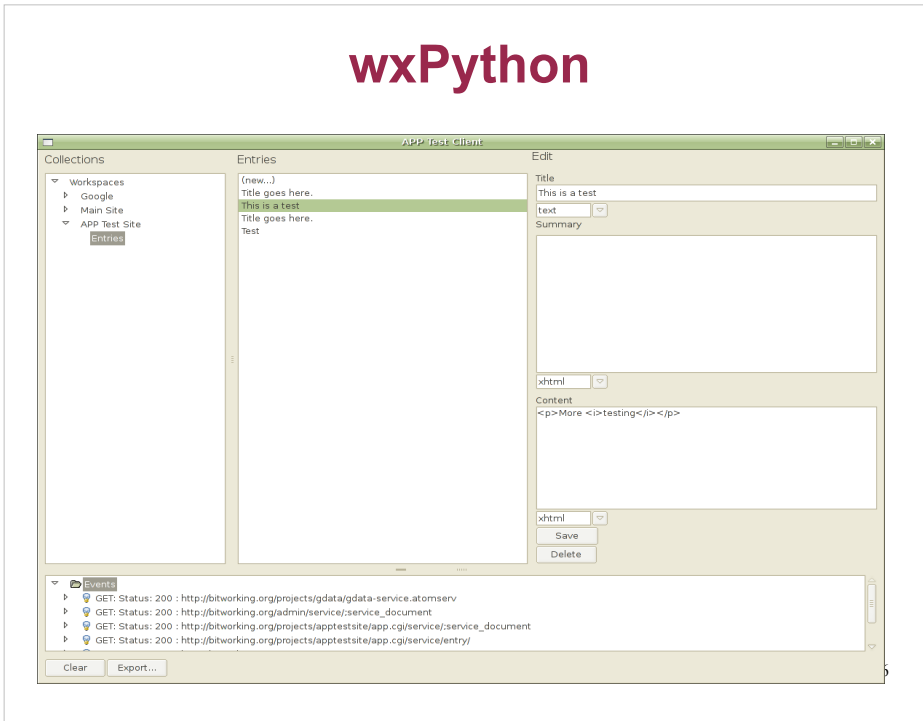
# The Matrix

	A.	B.	C.	D.	E.	F.	G.	H.	I.	J.	K.	L.	M.	N.	O.	P.
	Abdera (Snell)	Amplia (Peterson)	Blogger (Clinton)	AOL (Panzer)	Bligs (Snell)	Dogear (Snell)	Gregorio (Mehta)	Oracle (Reese)	Six (Green)	O'Reilly (Torres)	WordPres (Johnson)	Roller (Johnson)	Propono (Beyer)	PHP+DB2 (Beyer)	APP (Asakura)	O'Reilly (Deli)
1. APE (Bria)	loA1 (/)	loB1 X	loC1 X	loD1 (/)	loE1 (/)	loF1 (/)	loG1 (/)	loH1 (/)	loI1 -	loJ1 X	loK1 (/)	loL1 (/)	loM1 -	loN1 (/)	loO1 X	loP1 X
2. Abdera (Snell)	loA2 (/)	loB2 X	loC2 (/)	loD2 X	loE2 (/)	loF2 (/)	loG2 (/)	loH2 (/)	loI2 -	loJ2 (/)	loK2 (/)	loL2 (/)	loM2 -	loN2 (/)	loO2 X	loP2 (/)
3. O'Reilly (Fahlgren)	loA3 -	loB3 -	loC3 -	loD3 -	loE3 (/)	loF3 -	loG3 -	loH3 (/)	loI3 -	loJ3 (/)	loK3 -	loL3 -	loM3 -	loN3 (/)	loO3 -	loP3 -
4. Xamleon(Peterson)	loA4 -	loB4 -	loC4 -	loD4 -	loE4 -	loF4 -	loG4 -	loH4 -	loI4 -	loJ4 -	loK4 -	loL4 -	loM4 -	loN4 -	loO4 -	loP4 -
5. Gregorio	loA5 -	loB5 -	loC5 (/)	loD5 -	loE5 -	loF5 -	loG5 (/)	loH5 (/)	loI5 -	loJ5 -	loK5 (/)	loL5 -	loM5 -	loN5 (/)	loO5 -	loP5 -
6. Oracle (Mehta)	loA6 -	loB6 -	loC6 -	loD6 -	loE6 -	loF6 -	loG6 -	loH6 -	loI6 -	loJ6 -	loK6 -	loL6 -	loM6 -	loN6 (/)	loO6 -	loP6 -
7. Six Apart (Paschal)	loA7 -	loB7 -	loC7 X	loD7 -	loE7 -	loF7 -	loG7 X	loH7 -	loI7 -	loJ7 -	loK7 -	loL7 -	loM7 -	loN7 -	loO7 -	loP7 -
8. Windows Live Writer (Cheng)	loA8 -	loB8 -	loC8 (/)	loD8 -	loE8 (/)	loF8 X	loG8 (/)	loH8 (/)	loI8 -	loJ8 X	loK8 (/)	loL8 (/)	loM8 -	loN8 (/)	loO8 -	loP8 -
9. Flock (Loisant)	loA9 -	loB9 -	loC9 -	loD9 -	loE9 -	loF9 -	loG9 -	loH9 -	loI9 -	loJ9 -	loK9 -	loL9 -	loM9 -	loN9 -	loO9 -	loP9 -
10. Propono (Johnson)	loA10 -	loB10 -	loC10 (/)	loD10 -	loE10 (/)	loF10 -	loG10 X	loH10 X	loI10 -	loJ10 -	loK10 (/)	loL10 (/)	loM10 (/)	loN10 -	loO10 -	loP10 -
11. Nukular (McCulley)	loA11 -	loB11 (/)	loC11 (/)	loD11 X	loE11 -	loF11 -	loG11 (/)	loH11 (/)	loI11 -	loJ11 -	loK11 (/)	loL11 (/)	loM11 -	loN11 (/)	loO11 (/)	loP11 -
12. Javascript (Beyer)	loA12 -	loB12 -	loC12 -	loD12 -	loE12 -	loF12 -	loG12 -	loH12 -	loI12 -	loJ12 -	loK12 -	loL12 -	loM12 -	loN12 -	loO12 -	loP12 -
13. C# (Asakura)	loA13 -	loB13 -	loC13 -	loD13 -	loE13 -	loF13 -	loG13 X	loH13 -	loI13 X	loJ13 X	loK13 -	loL13 -	loM13 -	loN13 -	loO13 -	loP13 (/)
14. WSO2 (Marsh)	loA14 -	loB14 -	loC14 (/)	loD14 -	loE14 -	loF14 -	loG14 -	loH14 -	loI14 -	loJ14 -	loK14 -	loL14 -	loM14 -	loN14 -	loO14 -	loP14 -
15. Google data py	loA15 -	loB15 -	loC15 (/)	loD15 -	loE15 -	loF15 -	loG15 (/)	loH15 -	loI15 -	loJ15 -	loK15 -	loL15 -	loM15 -	loN15 (/)	loO15 -	loP15 -

This is the interop grid from April 2007. The rows are client and the columns are server implementations. The rows and columns highlighted in orange are open source implementations.

Open source is not a prerequisite for moving quickly, for example Joe Cheng of MS was there with Windows LiveWriter.

# wxPython



Not only did validation but also produced some debugging info. (so actually a hybrid between the two classes of software were talking about)

## URI Templates

```
http://example.org/{userid}
```

```
userid = 'fred'
```

```
http://example.org/fred
```

27

URI Templates are another area that I am working in. In this case the subject area is easier because it's not a protocol. The test cases are included in the spec, copied from the Python implementation.

Those unit tests have propagated with the ported implementations.

Now implemented in Python, PHP, Java (2), JavaScript, Ruby (2), and Erlang.

## Why it matters

Why you should get involved?

Better specs  
Better software  
Less time arguing

28

IETF 'running code' exists for a reason, some problems only come to light when you go to implement.

The dangling Else additions.

Feedback from hands on experience carries a lot of weight.

IPR