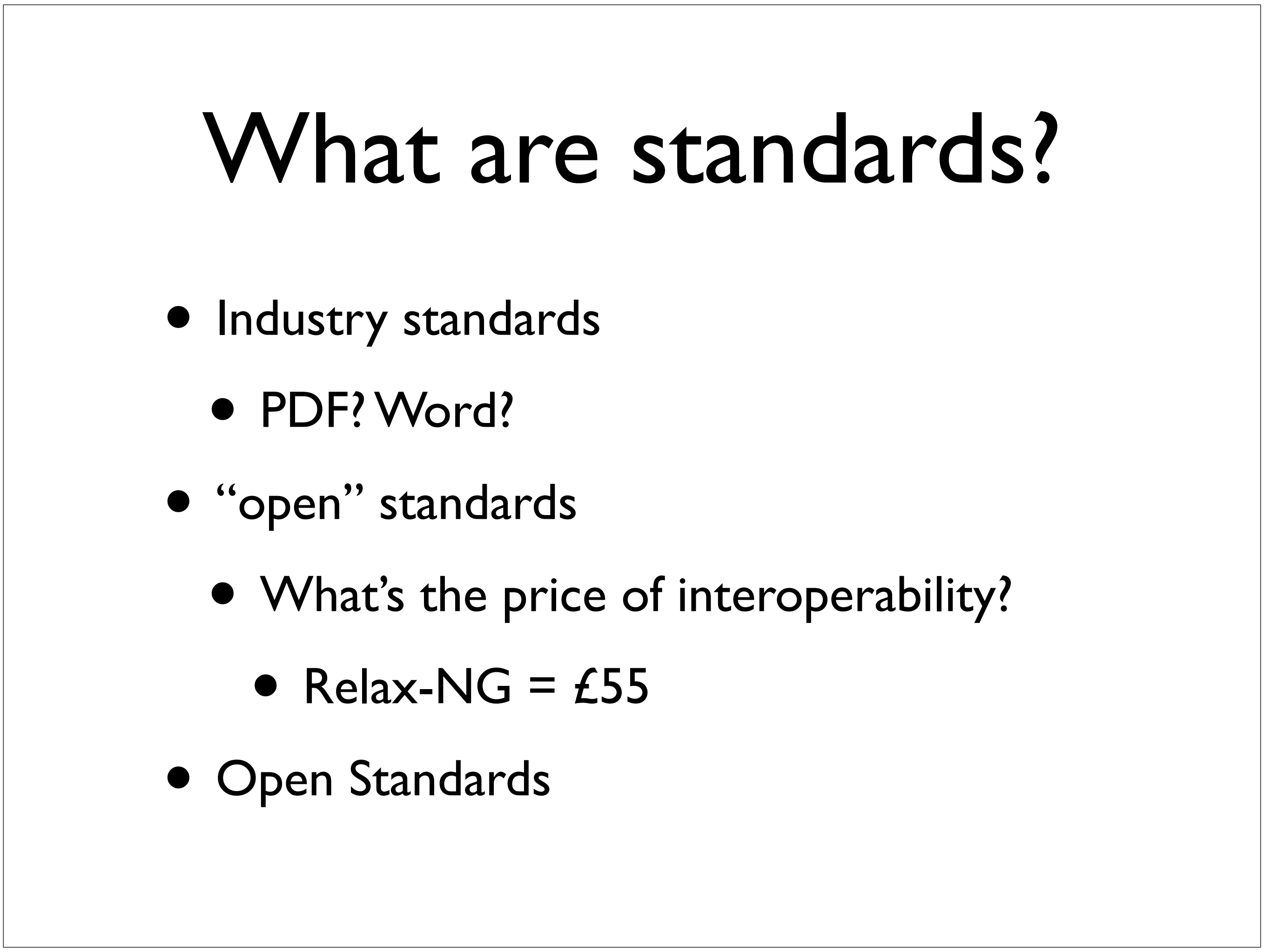
Open Standards in Open Source

Andrew Savory, Luminas

This is a tongue-in-cheek look at the symbiotic relationship between Open Standards and Open Source. It is designed to stimulate discussion rather than to be entirely truthful or accurate!

Introduction

- Andrew Savory:
  - OS developer for ~ 10 years
  - Developer on Apache Cocoon, Jackrabbit
  - Open Source pragmatist
  - Director of Luminas and Orixo



What are standards?

- Industry standards
  - PDF? Word?
- “open” standards
  - What’s the price of interoperability?
    - Relax-NG = £55
- Open Standards

The word “standard” is frequently abused, and there are several different types of “standard” in the IT industry:
- Industry standards: where the market-leader uses / abuses their position to push one way of working (typically file formats) and will rarely publish the specifications for that widely-adopted “standard”.
- “open” standards, which pretend to be widely available but where you have to pay the standards body to access the specifications.
- Open Standards, often in the form of Recommendations (W3C) or RFQs (IETF). These are designed by experts and made available to anyone, with feedback and improvements encouraged and expected.

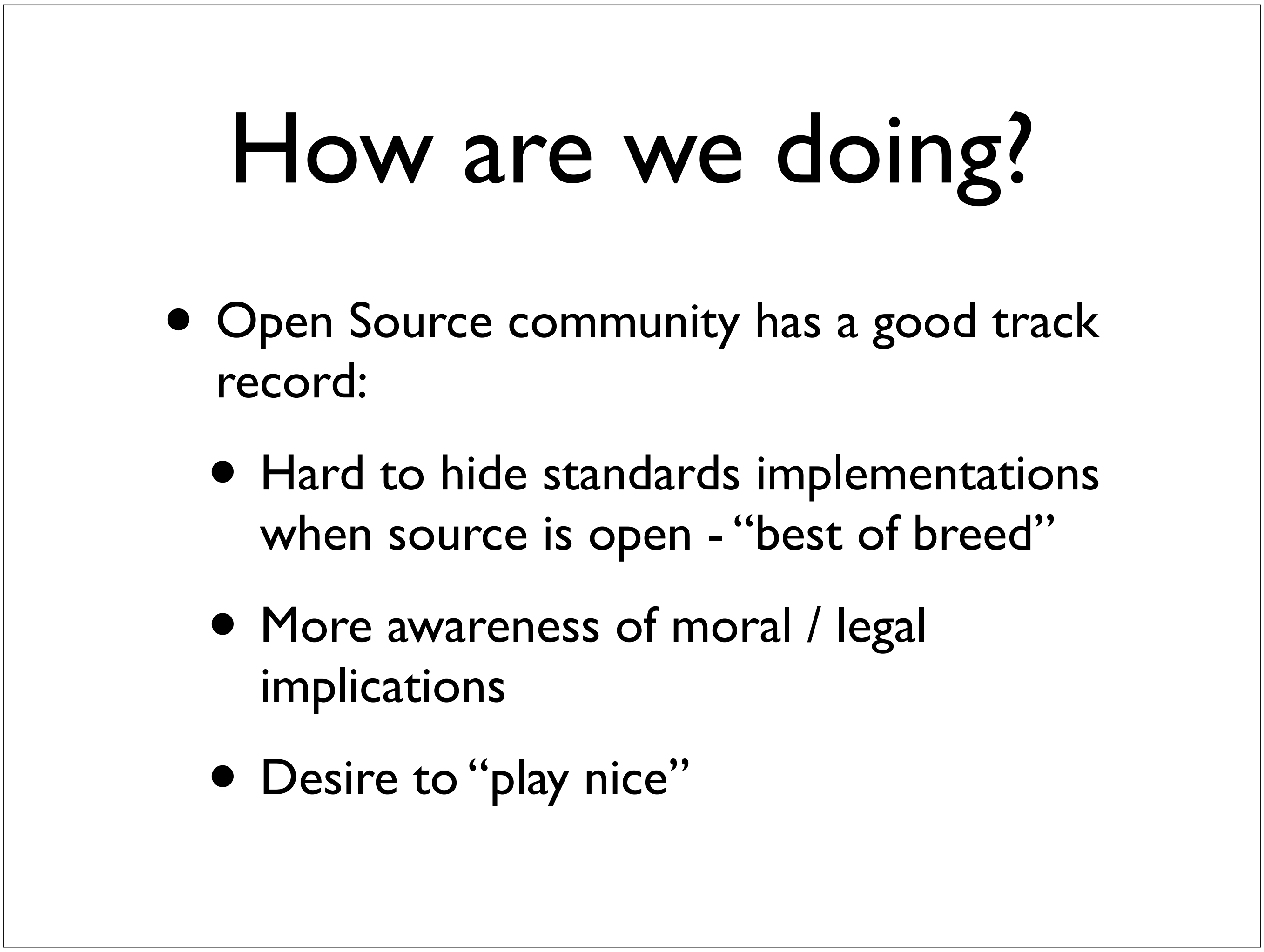
Why Standards?

- Interoperability
- Competition
- Security and testing

Why are standards important to Open Source developers?
Interoperability: we aren't interested in vendor lock-in. We want to make sure our software works with others.
- even some proprietary developers understand this: a good example is Brent Simmons, author of NetNewsWire

Competition: we're a fiercely competitive lot, and we each believe we're going to produce the best implementation.
Because Open Source developers love to compete with each other, we need a frame of reference - standards set out the ground rules.

Security and testing: how do we know we've done it properly? Do you fix 10 different implementations or write your standards in a secure way? How do you know the program does what it should? The standards tell you so.

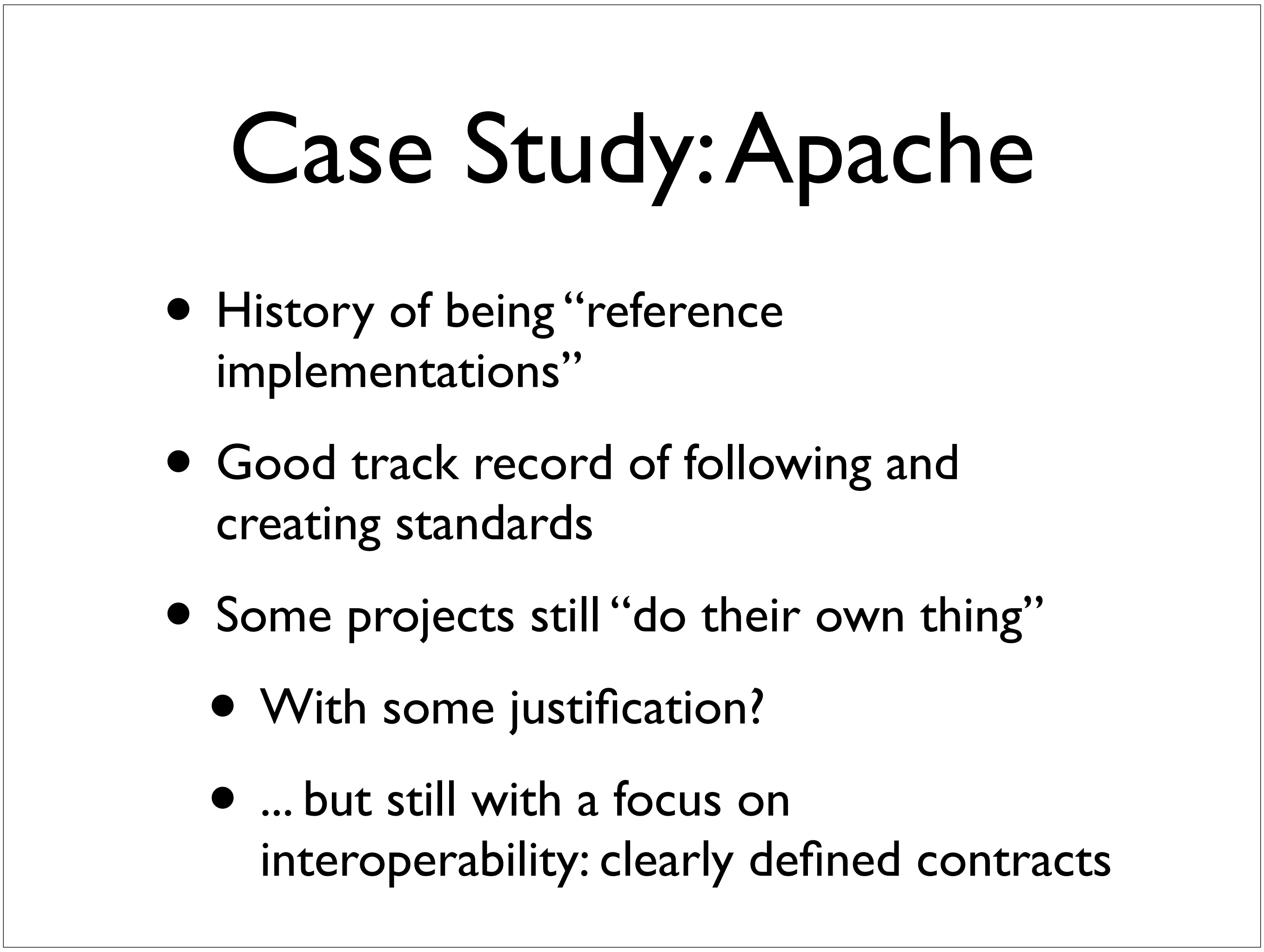


How are we doing?

- Open Source community has a good track record:
  - Hard to hide standards implementations when source is open - “best of breed”
  - More awareness of moral / legal implications
  - Desire to “play nice”

The open source community are big fans of standards, and tend to implement them whenever possible.

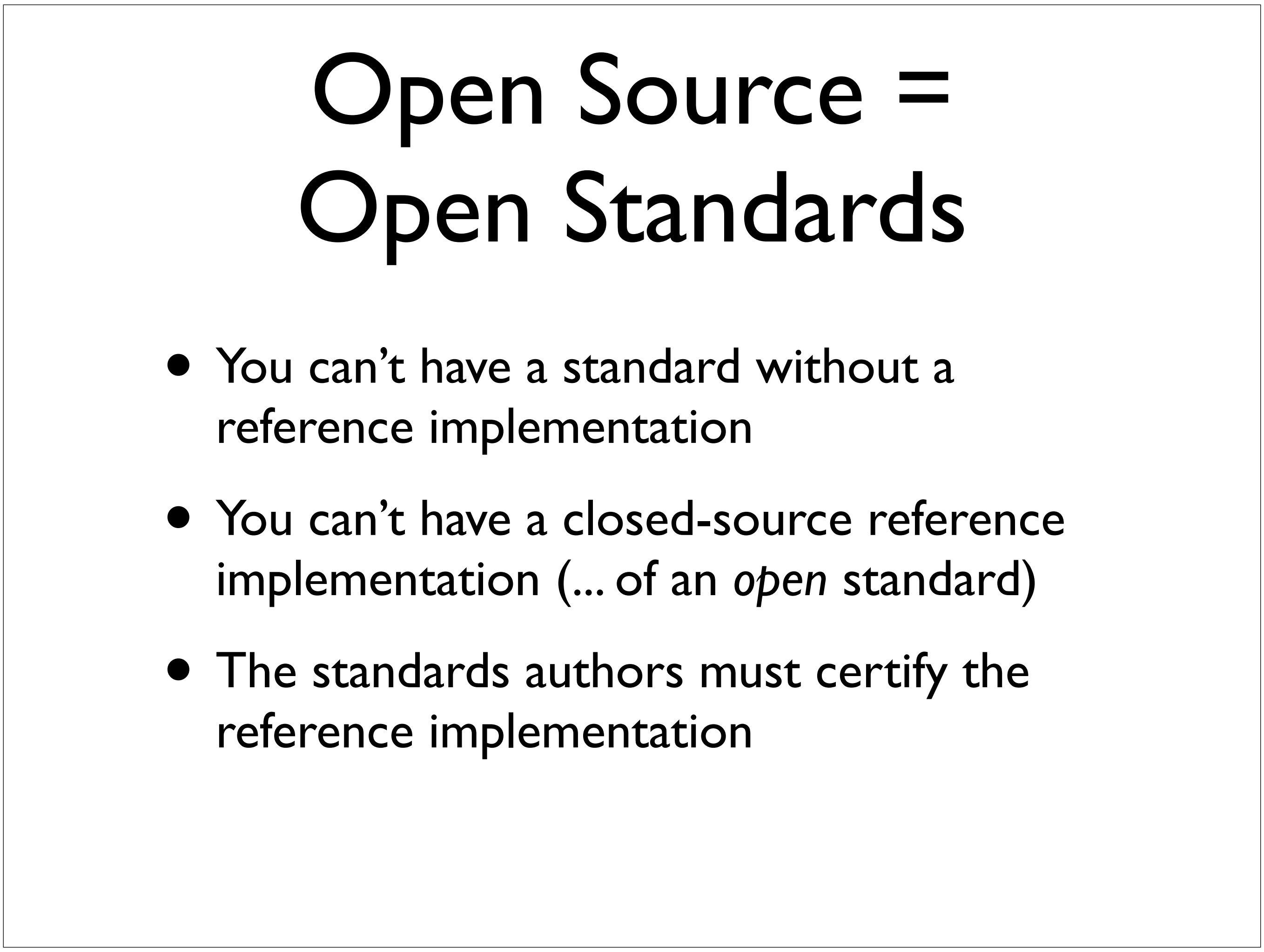
In return, we are good for the survival of a standard:
* we make sure the best possible implementation of a standard wins
* we help make sure standards are fair and implementable (preventing things like RAND)
* we work on standards because we love to play nice, and have no need to lock-in. We’re actively working to refine standards and software.



Case Study: Apache

- History of being "reference implementations"
- Good track record of following and creating standards
- Some projects still "do their own thing"
  - With some justification?
  - ... but still with a focus on interoperability: clearly defined contracts

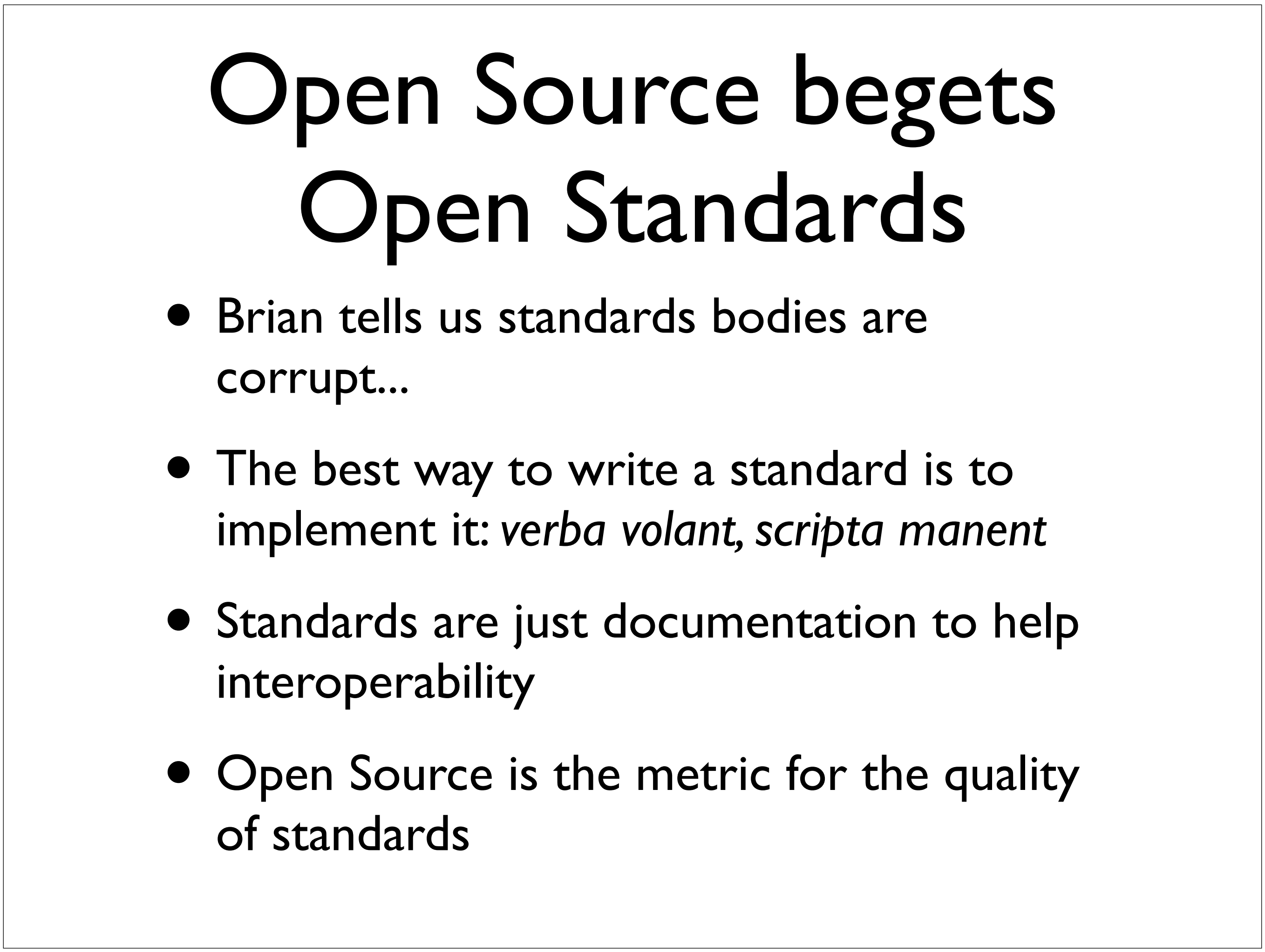
Apache is a good example of an OS developer community
Many example implementations of open standards can be found in Apache: Tomcat (Servlet Container), Pluto (Portals), Jackrabbit (Content Repository API).
Where a standard exists, it is often used. Where no suitable standard exists, efforts are made to create them: many ASF guys on JSR 168, JSR 170
Sometimes they diverge: in Cocoon, XForms dumped in favour of CForms, JSP dumped in favour of XSP. Often this is because the standard is considered incomplete, immature, or unsuitable



Open Source = Open Standards

- You can’t have a standard without a reference implementation
- You can’t have a closed-source reference implementation (... of an *open* standard)
- The standards authors must certify the reference implementation

Without open source there would be no (accepted) open standards.
The only way to ensure people implement a standard correctly is to show them what you mean.
If you’re going to show them ... you have to make it public to ensure everyone plays along.
You need a rubber-stamp on your standard: this is something the java community process gets right.
... W3C should get in on the act?
... this is what is good about Java? J2EE certification.



Open Source begets Open Standards

- Brian tells us standards bodies are corrupt...
- The best way to write a standard is to implement it: *verba volant, scripta manent*
- Standards are just documentation to help interoperability
- Open Source is the metric for the quality of standards

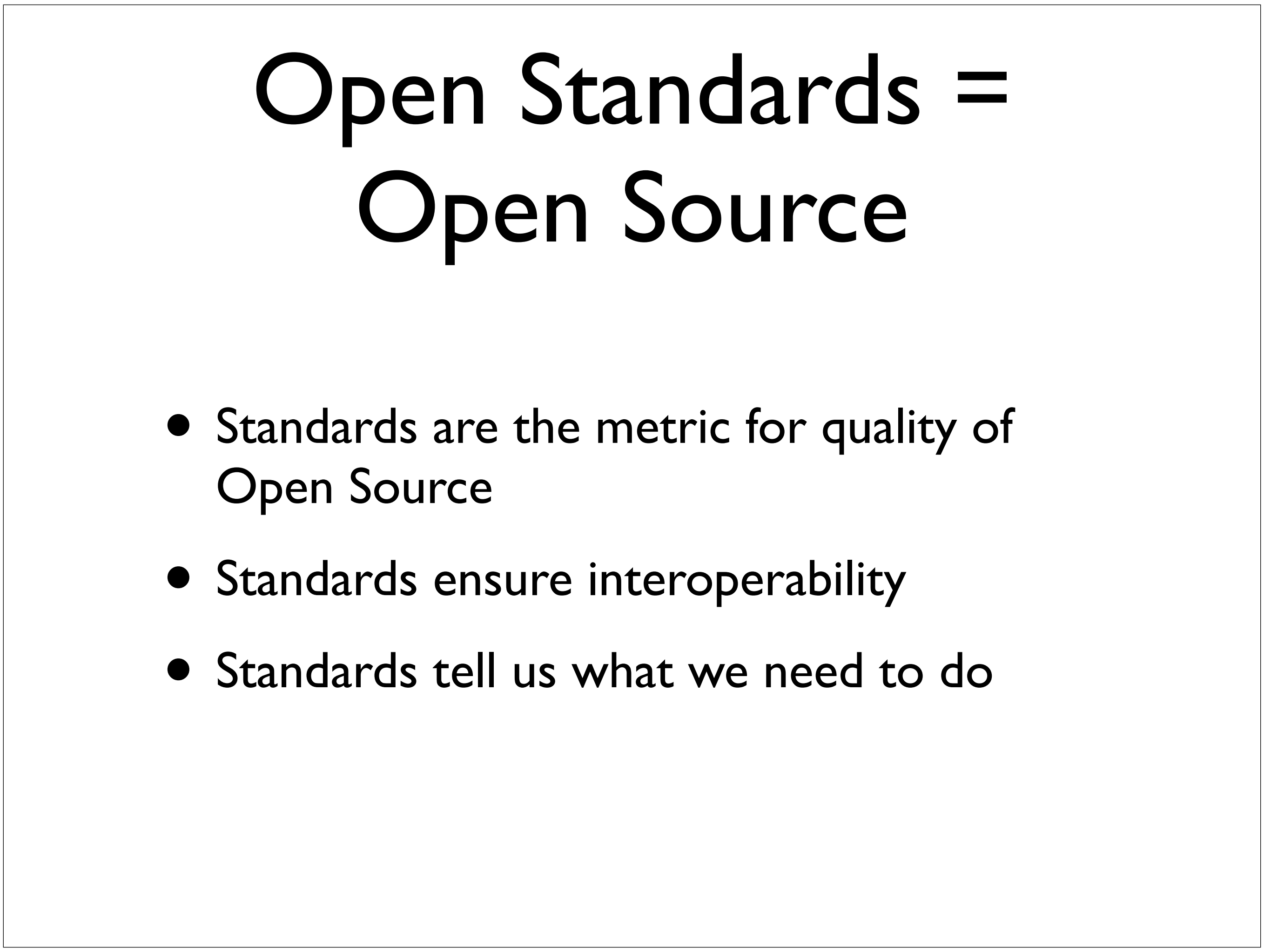
Which comes first, the chicken or the egg?
Groovy to JCR 241, ...?
So many standards are written with no consideration of the implementation:
Norman Walsh writes “XML 1.1: Dead on Arrival”:

The working group leading RELAX NG through the ISO standardization process has ruled that “an XML [1.1] document…can never be valid against a RELAX NG schema.” I expect the W3C XML Schema working group to conclude similarly that XML 1.1 documents cannot be validated with XML Schema 1.0 or 1.1. ¶

From Nicola Ken, an Apache contributor:

```
          - verba volant, scripta manent -
   (discussions get forgotten, just code remains)
```

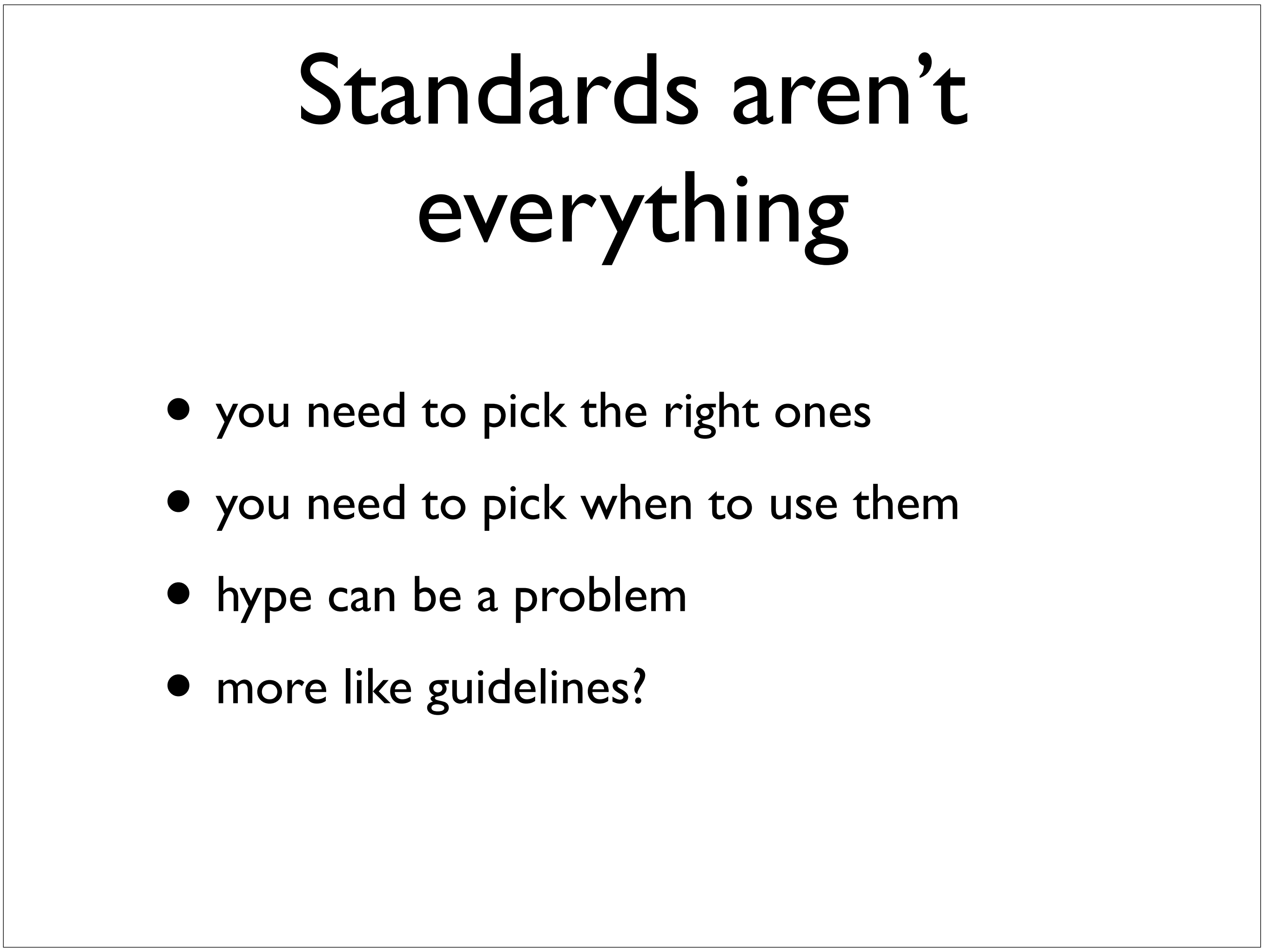
Open Source are saving the world from bad standards
– we can’t hide the hacks required to implement something



Open Standards = Open Source

- Standards are the metric for quality of Open Source
- Standards ensure interoperability
- Standards tell us what we need to do

Which comes first, the chicken or the egg?
We couldn’t write (some) software without a standard:
- how do you talk to a web server? (HTTP 1.1 RFC 2616)
Good open source implements a standard completely - or documents where it doesn’t (standards tell us what we still have left to do)



Standards aren't everything

- you need to pick the right ones
- you need to pick when to use them
- hype can be a problem
- more like guidelines?

GNUCash used XML standard for file format ...
... but still couldn't interoperate with it
... should have looked for Open Financial eXchange support
Daisy could have used WebDAV, JCR, ...
... but wouldn't have been finished if it did...
JSR 170 (JCR) seen as the silver bullet for interop between Open Source and proprietary ...
... finally we can get our content back out of Vignette, Documentum, etc
... but only if the big vendors implement it ... ooops.
Regardless of standards, you need to be:
* liberal in what you accept
* strict in what you transmit