



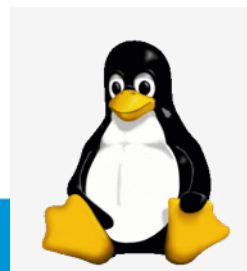
# In re: Standards vs Open Source

## Eyewitness Testimony

Mike Dolan  
The Linux Foundation  
25 May 2016

# The Linux Foundation

- Home to many of the world's most important open source and open standards projects across segments including operating systems, embedded, cloud, storage, networking, telecom platforms, internet of things and application platforms.



# The Linux Foundation

- Home to many of the world's most important open source and open standards projects across segments including operating systems, embedded, cloud, storage, networking, telecom platforms, internet of things and application platforms.



Every day, developer communities work on and contribute code to open source projects to build the best solutions possible for a particular objective

Contributions are voluntary, not compulsory and tend to “scratch their own itch”

# Linux kernel project: **scale of collaboration**

One release of the Linux kernel in February, 2015:

- 53,000 files
- 21,150,000 lines of source code
- 3,974 developers
- 440 companies

# Linux kernel project: **rate of development is increasing**

1 year of Linux kernel development (April 2015 to March 2016)

- 10,800 lines of code added
- 5,300 lines of code removed
- 1,875 lines of code modified
- 10+ older kernel version bug patches

# Linux kernel project: **rate of development is increasing**

1 year of Linux kernel development (April 2015 to March 2016)

- 10,800 lines of code added
- 5,300 lines of code removed
- 1,875 lines of code modified
- 10+ older kernel version bug patches

# Every Day

# “Open source development” according to standards professionals



By Courier Litho. Co., Buffalo, N.Y. - This image is available from the United States Library of Congress's Prints and Photographs division under the digital ID cph.3h00057.

License: Public Domain, <https://commons.wikimedia.org/w/index.php?curid=2037679>



# “Standards development” according to open source software professionals...



Source: [https://en.wikipedia.org/wiki/Common\\_box\\_turtle#/media/File:Florida\\_Box\\_Turtle\\_Digon3a.jpg](https://en.wikipedia.org/wiki/Common_box_turtle#/media/File:Florida_Box_Turtle_Digon3a.jpg)

License: CC BY-SA 3.0

# Open Source projects need standards, standards need open source implementations

- Many open source projects implement standard protocols
- Many standards are implemented using open source projects
- The standards development cycle needs feedback from development/testing
- The open source community needs agreement on certain deployment options (green or red)
- Open source projects welcome architecture direction (often resolved in standards bodies)

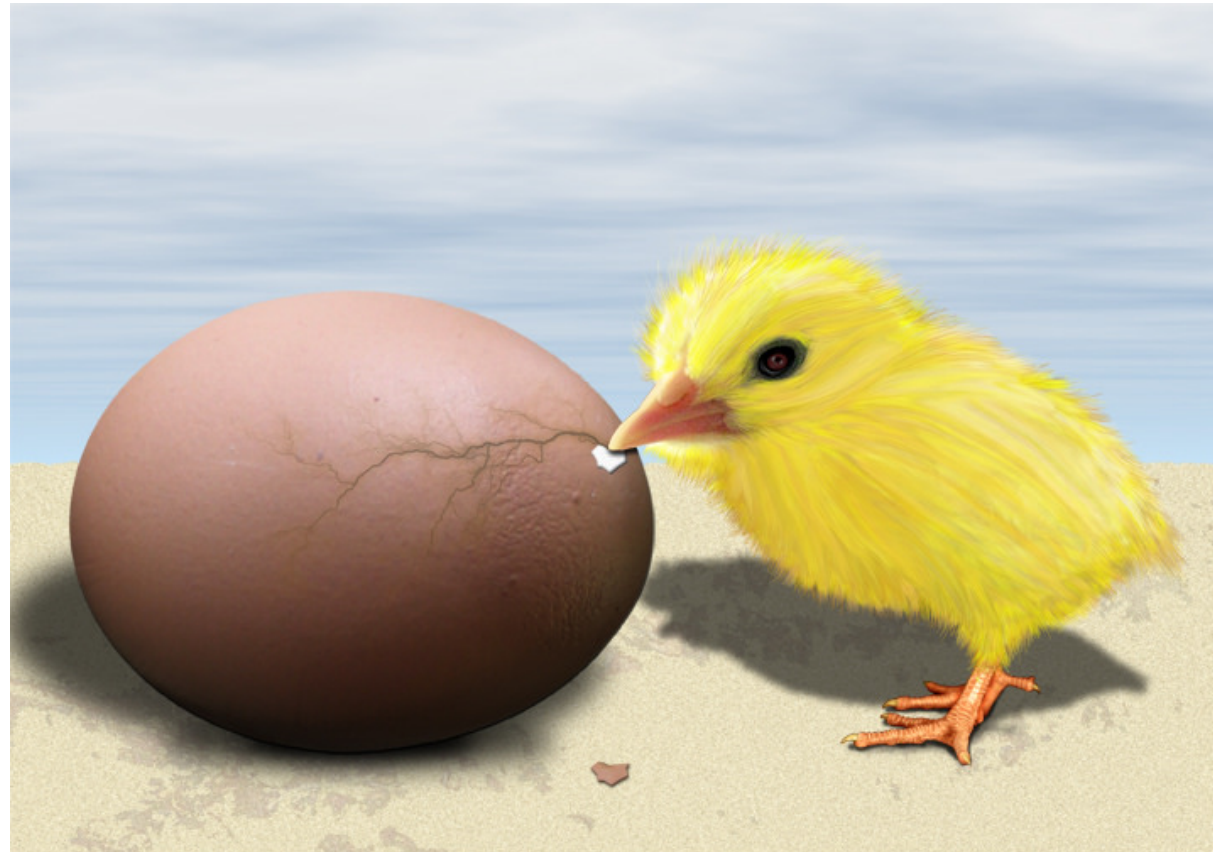
# Reality: SDOs and open source projects have similar characteristics

- Broad participation
- Clarity and control around intellectual property commitments and terms for usage
- Shared dependence / vulnerability on results
- User confidence in interoperability / compatibility
- Participants may not like each other and often compete

# Reality: the activities of standards and open source creation do have differences

- Scope: clearly defined vs evolving
- Time: rapid, iterative vs slow, deliberate
- Confidentiality: confidential vs open
- Sequence: code vs spec
- Participation: closed vs open
- Developer-driven vs Architect-driven

# The Great Debate



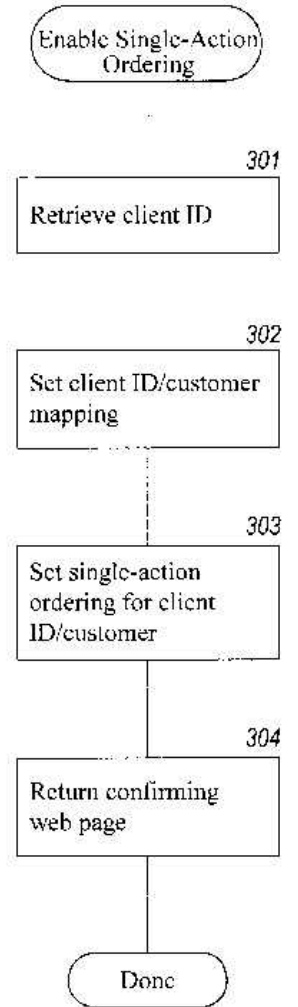
Source: <https://www.flickr.com/photos/donkeyhotey/13927746182>  
License: CC BY 2.0 <https://creativecommons.org/licenses/by/2.0/>

# Reality: communities will often eventually create both

- Open source projects that want interoperability across production deployments end up testing / certifying products and solutions to a set of interfaces and/or behaviors.
- Standards projects need to test their efforts in real deployments

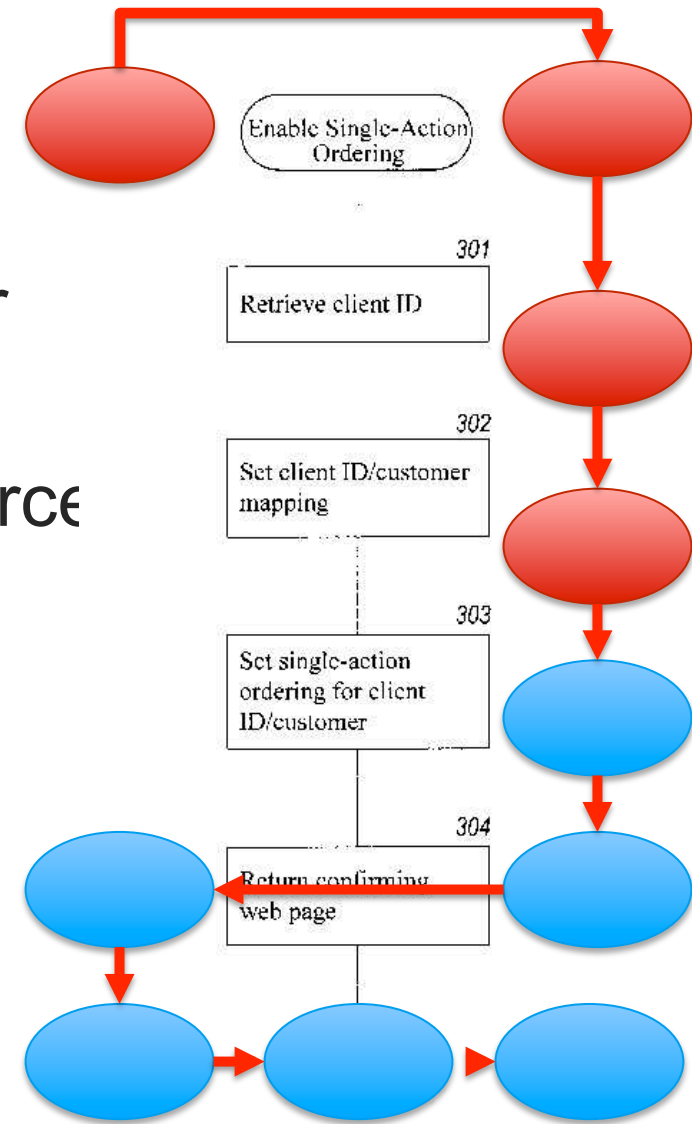
# Opinion: lawyers are spending inordinate time on the intersection of patents and open source

- Open source started as a backlash against copyright; most early project developers never considered patents



# Opinion: lawyers are spending inordinate time on the intersection of patents and open source

- Open source started as a backlash against copyright; most early project developers never considered patents
- Why? Patent claims implemented in open source software have a very different lifespan than patent claims implemented in hardware or proprietary software
- Open source software is often highly modular, pluggable and “changeable”.





# Issues we've seen arise...

- Does the scope of patent terms cover just what is contributed or the software release?
- Does the IP Policy conflict with the open source license
- Should both activities be under the same entity?
- “Hard wiring” open source project to a standard development process
- Too many barriers to contribution or participation will cause developers to go somewhere else
- Forking code bases happens
- Capturing patent license prior to sale
- Debates on patent grants vs non-assertion / defensive termination
- Developers like open mailing lists and open conversation, not confidential discussions they cannot participate in, “with enough eyes, all bugs are shallow” (including patent claims)

# Challenges with “FRAND open source software”

- Almost guaranteed a developer will just patch around your patented claims if they discover what they are
- Projects are often building/distributing binaries, if for no other reason than to test builds
- Contributory infringement? Who can distribute the code/binary openly?
- Forking of the project? GitHub forks by default for any work...
- Royalty Free FRAND (RAND-RF/RAND-Z) does not provide for an automatic license – companies still have to negotiate a license; royalty free license grants are a far more acceptable approach to open source communities
- Royalty Free FRAND often requires confidentiality which is challenging in an open participation community
- FRANDOSS is a challenging acronym ;-)

# How communities are self-remedying IP concerns

- Some projects include additional grants beyond the source code license to cover implementations of specifications
- Many projects are adding “architecture” committees, sub-projects to help define structure and approaches to create an internal standard
- Some are trying to do both standards and open source in same body... with a degree of challenge
- Most open source projects develop their code independent of the standards body activity, loose alignment with the standard
- Greater coordination of feedback from open implementations to the standards community
- Common / shared dependency on the project is often a natural deterrent; issues more likely to arise from NPEs.

# An example of doing both



- Source code for runc container runtime developed as a traditional open source project.
- Source code licensed under the Apache 2 with patent grant on contributions.
- Members in the project take on additional commitments regarding the specification for alternative implementations using the Open Web Foundation OWFa 1.0 (patent only) license.
- Prior to each major version software release, the members have 30 days to review before being committed to OWFa 1.0 (patent only).

## An example of doing both



- Source code for AllJoyn IoT framework developed as a traditional open source project.
- Source code licensed under the ISC license with no patent terms defined.
- Members and contributors to the project take on additional non-assertion and defensive termination commitments for the Board approved implementation.
- Prior to each major version software release, the members have 30 days to review before being committed to non-assertion against the full release (not just their commitments). Everyone who certifies their product against a compliant implementation gets the additional patent pledge benefits.

An example of doing both

OPEN  
CONNECTIVITY  
FOUNDATION™

+

IoTivity

- IoT specification standard developed as a standards project in Open Connectivity Foundation. Open source implementation IoTivity sponsored by OCF developed as an open source project under The Linux Foundation.
- IoTivity source code licensed under the Apache license with patent grant on contributions.
- OCF Members in the OCF project give and receive royalty free grants against the scope of the approved OCF specification, implemented in IoTivity.
- Strive for strong feedback loop between standard and open source development activities.

# Thank You!

Mike Dolan

[mdolan@linuxfoundation.org](mailto:mdolan@linuxfoundation.org)