

# Open Source Project Communities

**Prof. Dr. Dirk Riehle**

**Friedrich-Alexander University Erlangen-Nürnberg**

**FLOSS B04**

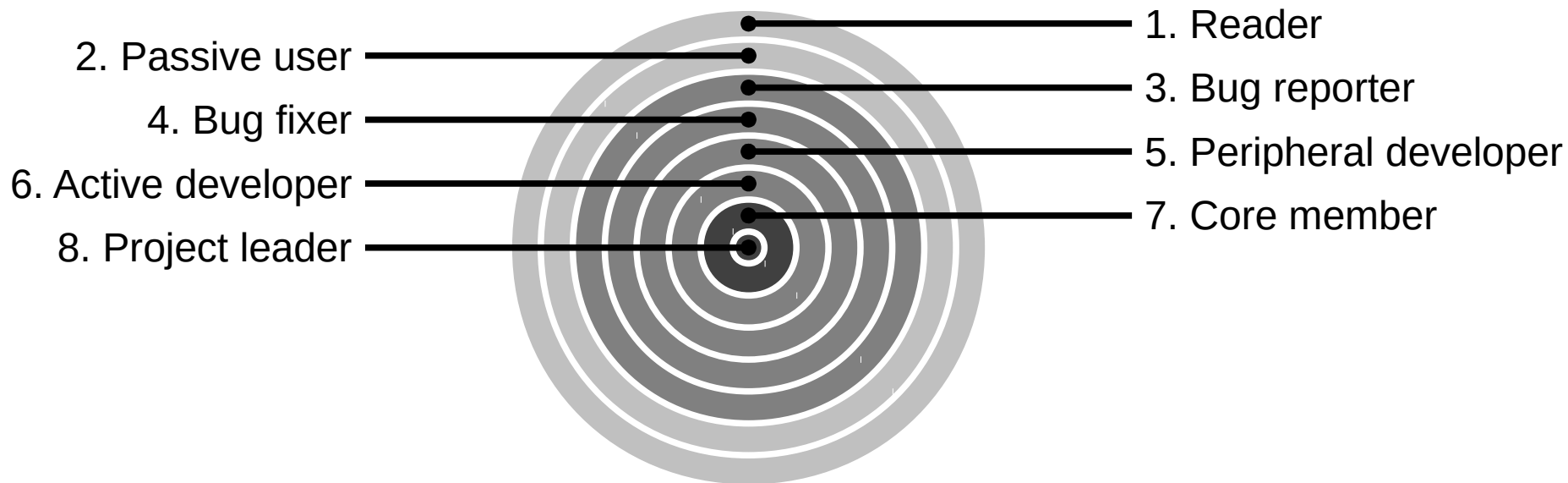
1. Legal innovation
2. **Process innovation**
3. Software tool innovation
4. Business model innovation

- **Project communities**
- Engineering processes

# Community Open Source Project

- Project is owned and managed by community
  - The project typically has only one license
  - The governing body has no direct commercial interests
  - No single vendor has undue / dominant influence

# The Onion Model [CH11]

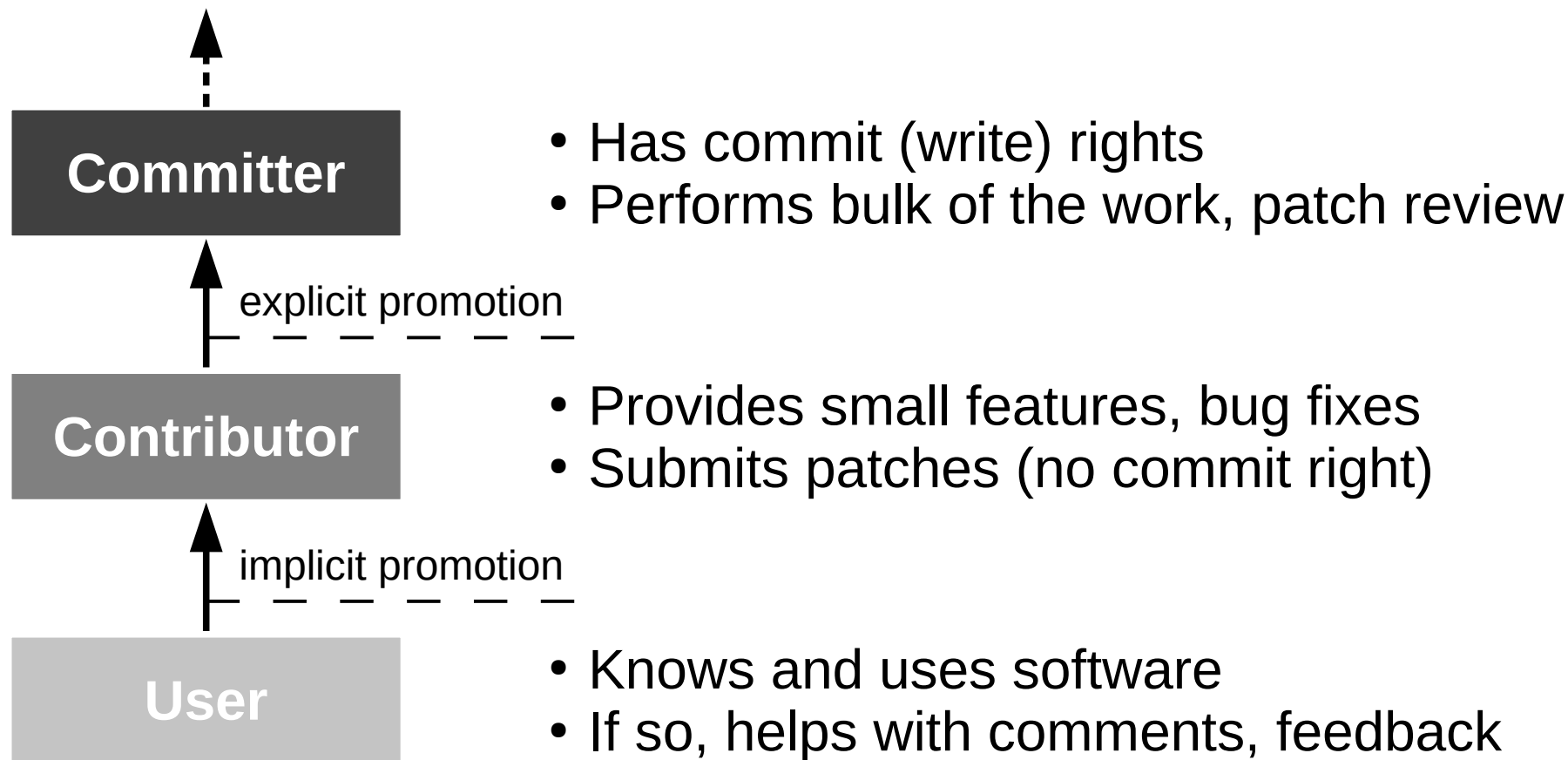


User

Contributor

Committer

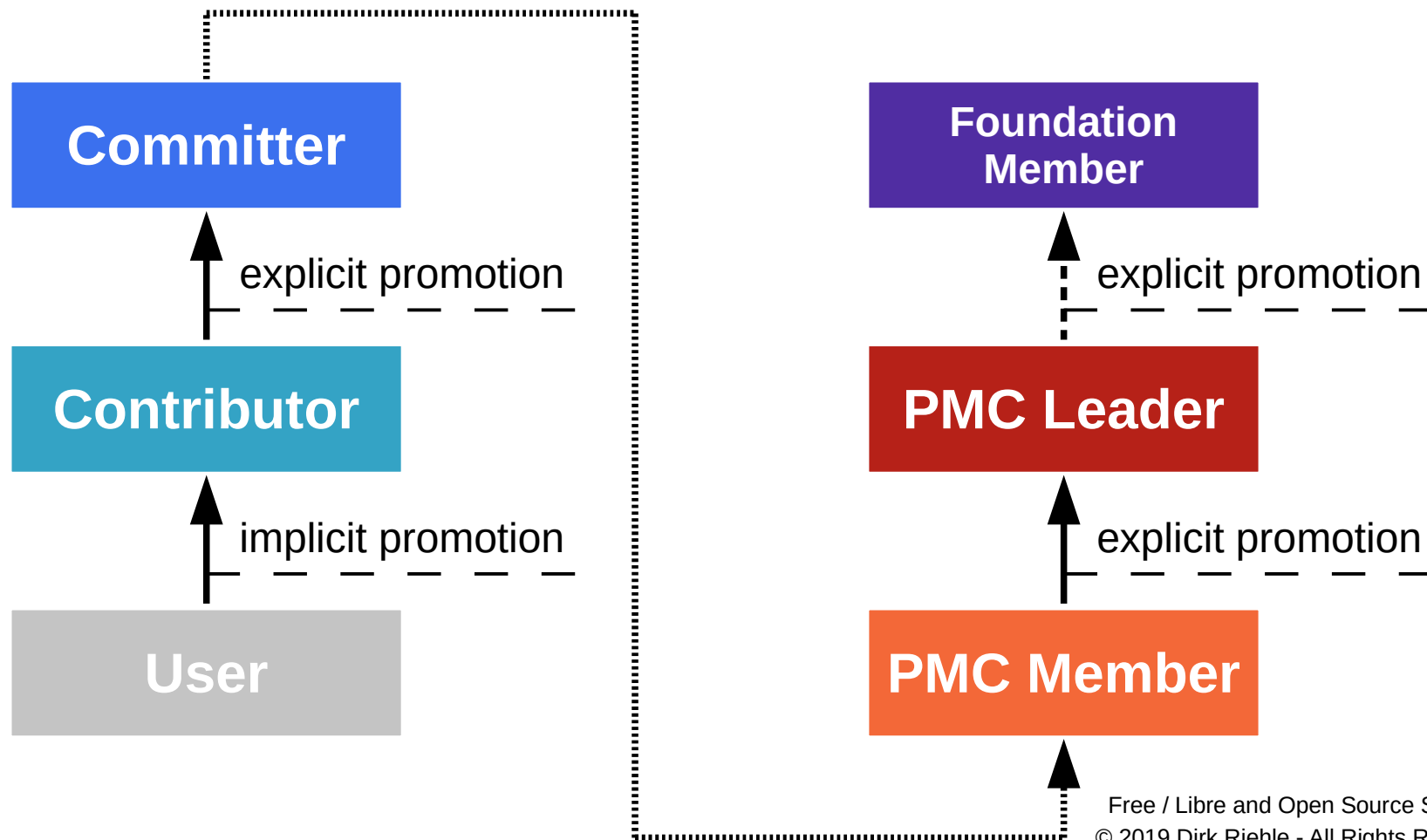
# The Basic Career Path [R15a]



# The Extended Career Path (Foundations)



# The Full Open Source Career Path





# The Apache Way 1 / 2 [A10b] (Official)

- Collaborative software development
- Commercial-friendly standard license
- Consistently high quality software
- Respectful, honest, technical-based interaction
- Faithful implementation of standards
- Security as a mandatory feature

# The Apache Way 2 / 2 [C17] (Curcuru)

- Charity
  - Apache's mission is providing software for the public good
- Community
  - Many of us are more effective than all of us
- Consensus
  - Getting good enough consensus is often better than voting
- Merit
  - Those that have proven they can do, get to do more
- Open
  - Technical decisions are discussed, decided, and archived publicly
- Pragmatic
  - Apache projects use the broadly permissive Apache license

# The 3-4-5 Model of Project Communities

1. **Three principles** open collaboration
2. **Four practices** of open communication
3. **Five stages** of project volunteering

# The Three Principles of Open Collaboration

1. **Egalitarian** participation
2. **Meritocratic** decision-making
3. **Self-organizing** processes

- **Traditional Work**

- **Hierarchical project assignment**
  - Closed and hidden silos
  - Assigned to project
- **Status-oriented decision making**
  - Public + private discussions
  - Hierarchical status decides
- **Predefined processes**
  - Prescribed process
  - Prescribed jobs

- **Open Collaboration**

- **Egalitarian participation**
  - Open for contribution
  - Everyone can contribute
- **Meritocratic decision-making**
  - Public discussion process
  - Decisions based on merit
- **Self-organizing processes**
  - People find their process
  - People find their project

# The Four Practices of Open Communication

- 1. Public**
- 2. Written**
- 3. Complete**
- 4. Archived**

- Public communication
  - Is visible to everyone
  - Creates trust and credibility
  - Does not a priori exclude anyone
- Almost all communication should be public
  - Tools: mailing lists, forums, wikis, etc.
  - Avoid private emails, phone conversations
  - Summarize private discussions using email

# Written Communication 2 / 4

- Written communication
  - Increases the reach of communication
  - Reduces barriers to participation (time and location)
  - Supports archiving of discussions
- (Almost) all communication should be in written form
  - Creates transparency, traceability, history
  - Reduces barriers to communication
  - Makes it easier to gain volunteers
- “If it wasn't written down, it might as well not exist.” [F05]



# Complete Communication 3 / 4

- Complete communication
  - Increases effectiveness of communication
  - Supports the on-boarding process
  - Creates more trust, credibility

- Archived communication
  - Simplifies documentation
  - Supports on-boarding process
  - Creates yet more trust, credibility

# Project Immigration [B11]

- User

1. Needs to get a job done
2. Searches web for software
3. Finds a matching project
4. Checks out the project
5. Gives it a try; is happy
6. Finds a bug; reports it
7. **Helps replicate bug; triage**
8. Engages in conversation
9. **Helps other users**
10. **Contributes to docs**
11. Leads project

- Developer

1. Needs to solve a problem
2. Searches web for software
3. Finds a matching project
4. Checks out the project
5. Gives it a try; is happy
6. Finds a bug; reports it
7. **Fixes bug; submits patch**
8. Engages in conversation
9. **Keeps finding + fixing bugs**
10. **Receives commit rights**
11. Leads project

# The Five Stages of Volunteering [R15b]

#	Volunteer View	Project View
1	Find project	Market project
2	Understand project	Explain project
3	Engage with project	Engage with volunteer
4	Work within project	Work with contributor
5	Lead project	Enable career

# Definition of Best Practice

- A best practice is a broadly-accepted, often informally-defined method for achieving a specific goal. (Cf. State-of-the-art.) [DR]
- Best practices in this presentation are marked as a reference if they were derived from related work. My interpretation typically evolved the original practice, though.

# #1 Volunteer: Find Project

- Stumble (and find)
  - Stumble upon project
  - **Listen to word-of-mouth**
- Search (to find)
  - Use general search engine
  - **Search on open source portal**

# #1 Project: Market Project 1 / 2

- Active outreach [F05]
  - Choose a good project name
  - Define all relevant channels
  - Use channels consistently
  - Announce visibly
  - Be matter of fact

Home » Foundation

About »


Projects

People »

Get Involved »

Download

Support Apache »



Google Custom Search

The Apache Way

Contribute

ASF Sponsors

Publicly archived mailing lists are critical to the operation of the Apache Software Foundation (ASF) and to our many Apache Projects. Apache projects use mailing lists to coordinate development of the software and administration of the organization. Mailing lists also serve as a primary support channel where users can help each other learn to use the software.

Mailing list participants are expected to abide by the well-established rules of "netiquette", as well as the ASF's published Code of Conduct. We also have [Tips for email contributors](#).

Please be aware that email sent to Apache developer or user mailing lists will be subject to our [Public Forum Archive Policy](#).

Note that some Apache mailing lists are not public; they are often but not always named **private@**. Be sure not to take emails from private discussions or mailing lists into a public forum or list unless there is agreement of all parties.

## SUBSCRIBING AND UNSUBSCRIBING

Unless otherwise noted, these lists are managed by [ezmin](#), which means that if you want to subscribe to [List@apache.org](#) then you need to send a message to [List-subscribe@apache.org](#).

To get off a list, send a message to [List-unsubscribe@apache.org](#).

To unsubscribe a different e-mail - e.g. you used to be subscribed as [user@oldname.example](#) - send a message to [List-unsubscribe-user=oldname.example@apache.org](#).

[The list moderators can also do this.]

If you do not recall the address you are subscribed as, view the full headers of any email you received from the list and examine the Return-Path, Received, and List-Id headers; these should embed both the address you are subscribed as and the address of the mailing list.

Note: Spam filters don't like messages with empty Subjects; just use e.g. "subscribe" or "unsubscribe". Spam filters are also more likely to reject HTML-formatted messages; please use plain text.

After you send the subscribe or unsubscribe request, the list manager will send you a confirmation e-mail in reply. You must reply to this e-mail to complete the process. If you have not received the confirmation request, check that it has not been marked as spam.

Note: private lists generally require a moderator to confirm the subscription. This may take a day or two. If necessary, try again after a few days. For this reason it's recommended to use the self-subscribe app instead (if you are a committer); that avoids needing to wait for the human moderators to check and green-light your subscription request.

Also check [this page](#) for information on how to contact the list owner (moderator), send a message to <https://www.apache.org/foundation/maillinglists.html>.

Hacker News

new | comments | show | ask | jobs | submit

login

Announcing Keyless SSL (cloudflare.com)

509 points by jgrahamc 529 days ago | past | web | 184 comments

lucble 529 days ago

For those who want to understand how it works (it took me a minute, so I'll try to explain it simpler):

In simplified terms, the server usually stores a public and private key, and sends the public key to the client. The client generates a random password, encrypts it with the server's public key, and sends it to the server. Only anyone with the private key can decrypt the message, and that should only be the server.

Now you don't want to hand over this private key to Cloudflare if you don't need to, because then they can read all traffic. Up until now, you needed to.

What they did was take the private key and move it to a keyserver, owned by your bank or whomever. Every time the Cloudflare server receives a random password (which is encrypted with the public key) it just asks the keyserver "what does this encrypted message say?" After that it has the password to the connection and can read what the client (the browser) is sending, and write data back over the same encrypted connection. Without ever knowing what the private key was.

The connection from Cloudflare to your bank's webserver and keyserver can be encrypted in whatever way. It could be a fixed key for AES, it could be another long-lasting TLS connection (the overhead is mostly in the connection setup)... this isn't the interesting part and can be solved in a hundred fine ways.

Edit: Removed my opinion from this post. Any downvotes for my opinion would also push the explanation down (which I hope is useful to some). I mostly agree with the other comments anyway.

Home

Moments

Notifications

Messages

Twitter

Search Twitter

934

85

33.8K

30

Following

Apache - The ASF

The official Twitter feed of The Apache Software Foundation [apache.org](#)

Joined December 2008

Tweet to Apache - The ASF

64 Followers you know

Photos and videos

Apache - The ASF

Want to go to @ApacheCon? We may be able to help get you there! Apply for Travel Assistance by 2 March [apache.org/travel/conference](#)

Apache - The ASF

The Apache News Round-up: week ending 26 Feb [s.apache.org/lysk](#) @ApacheGroovy @HBase @spwilk @apachekafka @SoitLucene @TheApacheTomcat

Apache - The ASF

Do you love Apache? We love you too! Join us @ApacheCon: Travel Assistance available --apply by 2 March [apache.org/travel/conference](#)

Apache - The ASF

The Apache Software Foundation Operations Summary: November 2015 - January 2016 [s.apache.org/9zLj](#) #Apache #Open #Innovation #Community

Apache - The ASF

Love Apache? Want to attend @ApacheCon? Need travel assistance? Hurry --applications close 2 March!

Who to follow

New Scientist

Science News

Festival de Cannes

Trends

Happy Birthday Justin Bieber

Adonian

KLHAY

Kuitham

Phanindar

HAPPIEST

Calli

George Kennedy

Social Service

TEEN WOLF IS OUR ANCHOR

THE LINUX FOUNDATION

日本語

Log in

LF Sites

HOME

TRAINING

EVENTS

COLLABORATIVE PROJECTS

Home

Program

Attend

Sponsors

Extend the Experience

Archive

CFP Dashboard

APACHECON NORTH AMERICA

May 11 - 13, 2016

Hyatt Regency Vancouver, Vancouver, BC

#ApacheCon

ApacheCon North America will bring together the open source community to learn about and collaborate on the technologies and projects driving the future of open source, web technologies and cloud computing.

Sponsor the Event

Register Now

Learn More

Venue

Hyatt Regency Vancouver

VANCOUVER, BC



# #1 Project: Market Project 2 / 2

- Passive inflow
  - Register on all relevant places
  - Support search engines
  - Support lucky chance
  - Use open source portal features
  - Provide findable summaries
  - Work to support portal metrics



## #2 Volunteer: Understand Project

- Read up on project

## #2 Project: Explain Project [F05]

- Have clear mission statement
- Provide examples and screenshots
- State license and terms
- Provide simple downloads
- Show current and future features
- Show development status
- Provide user documentation
- State whether volunteers are welcome
- State rules of engagement

# #3 Volunteer: Engage with Project 1 / 2 [B11]

- (Volunteer) **users**
  - Do your homework first
    - Search archives
    - Read documentation
  - Communicate prudently
    - Be matter of fact
    - Don't jump to conclusions

## #3 Volunteer: Engage with Project 2 / 2 [B11]

- (Volunteer) **developers**
  - Respect project practices
  - Respect project culture
  - Accept guidance

## [Geoserver-users] Unable to add MrSID as a layer

From: Ralph Dell <RDell@Ca...> - 2011-07-07 17:38

Attachments: [Message as HTML](#)

This is my first post to this list.

I am running geoserver 2.0.3, jetty, as a windows service. I have installed GDAL 1.7.3. GDAL\_DATA is set up, and the gdal native libraries are in GEOSERVER\_HOME/wrapper/lib. I have copied a MrSID to GEOSERVER\_DATA/data/MrSID folder which I created.

When adding a new store MrSID is one of my Raster Data Sources, I set the connection parameter to file:data/MrSID/Catawba\_West.sid and are able to add the store. I assume the next step is adding the mrsid as a layer, and here is the problem no layer shows up for that store, I'm confused. The coveragestore.xml looks like this

# How to Ask a Question 1 / 2

- The “simple” way [L11]
  - **I would like to do X (but it doesn't work)**
  - **I have tried Y to achieve X**
  - **Then Z happened**
- Additional information
  - configuration
  - version information
- Preferably no attachments
- Short email (30 lines)



# How to Ask a Question 2 / 2

- The “smart” way [RM14]
  - Choose your forum carefully
  - Stack Overflow
  - Web and IRC forums
  - As a second step, use project mailing lists
  - Use meaningful, specific subject headers
  - Make it easy to reply
  - Write in clear, grammatical, correctly-spelled language
  - Send questions in accessible, standard formats
  - Be precise and informative about your problem
  - Volume is not precision
  - Don't rush to claim that you have found a bug
  - Groveling is not a substitute for doing your homework
  - Describe the problem's symptoms, not your guesses
  - Describe your problem's symptoms in chronological order
  - Describe the goal, not the step
  - Don't ask people to reply by private e-mail
  - Be explicit about your question
  - When asking about code
  - Don't post homework questions
  - Prune pointless queries
  - Don't flag your question as “Urgent”, even if it is for you
  - Courtesy never hurts, and sometimes helps
  - Follow up with a brief note on the solution

# Basics of Email Communication

- OpenStack mailing list etiquette [O15]
  - Subjects
  - Formatting
  - Netiquette

# Effective Written Communication

- Every (email) contribution should have **one** purpose
  - Open a new issue that needs discussion
  - Support or argue against the issue at hand
  - Conclude discussion and summarize for archiving
- Follow patterns of effective writing
  - 72 characters, use subject line, use real sentences
  - Assume good faith, make no assumptions, use neutral tone
  - Terseness is acceptable, rudeness is not
  - Use URLs for reference, shortening
- Use the correct tool for the purpose
  - For example, no discussions in the bug tracker

# Archive as Public Documentation

- Reasons for public documentation of project
  - Public documentation creates transparency, credibility, trust
  - Credibility and trust helps get volunteer contribution
  - Good public documentation reduces work, supports scaling up
- Public documentation needs to be backed into process
  - Mailing lists should have publicly readable archive
  - Wiki should be crosslinked with mailing list, bug tracker, code repository
  - Documentation work should be publicly recognized as valuable
- What to document? Anything of value to the project

# Social Aspects of Communication [CF08]

- Be consistent in your communication
  - Adopt a real/serious name
  - Use that name consistently
  - Don't self-aggrandize
  - Use proper language
- Show no tolerance against rudeness
  - Rudeness destroys communities
  - Separate rudeness from content
- Companies have training programs



*"On the Internet, nobody knows you're a dog."*

# Advice on Email Communication

- Sue Gardner on Wiki[p/m]edia communications [G16]
  - “You are going to be doing a lot of writing. Do it well.”
  - “In all your communications, be conscious of your group’s diversity.”
  - “Be cautious about creating an insider culture.”
  - “Publish early, publish often.”
  - “Be aware that volunteer time is different from paid time.”
  - “Write (and publish) a greater volume of stuff than you think you should.”
  - “Assume good faith.”
  - “Bias towards transparency. Way, way, way more than you think you should.”
  - “When you change your mind, say it publicly, and explain why.”
  - “Pay attention to people you disagree with.”

## #3 Project: Engage with Volunteer 1 / 2 [F05]

- (Engage) with everyone
  - Show how to reach you
  - Welcome people to community
  - Stop any rudeness
  - Avoid private discussions
  - Accept initial redundancy
  - Provide simple tasks

## #3 Project: Engage with Volunteer 2 / 2 [F05]

- (Engage) with developers
  - Provide tool access
  - Show requirements list
  - Provide developer guidelines
  - Provide developer docs



# What Motivates Volunteers?

- Technical Motivation
  - To solve their technical problem
- Economic Motivation
  - To signal their capabilities to the labor market
  - To achieve status of economic value
- Social Motivation
  - To feel validated by other people
  - To enjoy a community
  - To do good

# Getting to the First Contribution

- Make first contribution easy
- Make contributions incremental
- Remove arbitrary tool obstacles
- Call out lurkers from shadows

## #4 Volunteer: Work within Project

- Be responsive
- Be reliable
- Contribute

## #4 Project: Work with Contributor 1 / 3 [F05]

- General practices
  - Be component-oriented
  - Work from features
  - No discussions in bug tracker

## #4 Project: Work with Contributor 2 / 3 [F05]

- **Working with a new contributor**
  - React speedily, don't sit on patches
  - Turn contributions into conversations
  - Practice conspicuous code review
  - Track contributions, provide credit
  - Praise plentiful, criticize specifically
  - Prevent territoriality

## #4 Project: Work with Contributor 3 / 3 [F05]

- **Working with a regular contributor**
  - Track volunteer interests, assign accordingly
  - Distinguish inquiry from assignment
  - Share technical as well as managerial tasks
  - Follow-up to delegated assignment
  - Document practices and traditions
  - Archive practice descriptions

# Meritocratic Decision Making Process

- Process for decision making depends on the project
  - Compare Linux Kernel vs. PostgreSQL vs. Tiki Wiki CMS
  - All involve some form of mailing list discussion
  - Sometimes, private side discussions are OK
- Most consensus-based projects avoid votes
  - Having to vote indicates a failure of the process
  - But if you have to ...
    - Idiosyncratic -1,  $\pm 0$ , +1 voting process [A10]
    - Votes are called on project roles, design decisions, resource allocations
    - Important example: Committer elections (determine social set-up)
- Decisions are summarized and archived

# Illustration of How Numbers Play Out

<b>Committers</b>	<b>1</b>
<b>Contributors</b>	<b>100</b>
<b>Users</b>	<b>10000</b>



**Time is your scarcest resource**

**Recruiting takes time**

**Therefore:**

**View recruiting as investment**

**Align time spent with return received**

**Recruit for the long-term [F05]**

# Committer Announcements



**Joe Conway** <mail@joeconway.com>

11:22 PM (10 hours ago) ☆



to Andres, Tom, pgsql-hackers ▾

On 06/01/2018 05:14 PM, Andres Freund wrote:

> On 2018-06-01 17:05:11 -0400, Tom Lane wrote:

>> The core team is pleased to announce the appointment of seven

>> new Postgres committers:

>>

>> Etsuro Fujita

>> Peter Geoghegan

>> Amit Kapila

>> Alexander Korotkov

>> Thomas Munro

>> Michael Paquier

>> Tomas Vondra

>>

>> Congratulations to all!

>

> +1!

+7!

\*\*\*

## #5 Volunteer: Lead Project 1 / 2

- Lead project
  - Take responsibility
  - Praise other contributors
  - Acknowledge contributions
  - Support towards their manager
  - Provide tokens of appreciation

## #5 Volunteer: Lead Project 2 / 2

- Enable community career
  - Define community career
  - Define roles and positions
  - Decide promotion privately
  - Announce promotion clearly

# Dealing with Difficult People 1 / 4 [CF08]

- Understand the problem
  - Difficult (“poisonous”) people
    - May waste your time
    - May split the project
    - May ruin the project

# Dealing with Difficult People 2 / 4

- Prepare for the problem
  - Build a healthy community
  - Document all decisions

# Dealing with Difficult People 3 / 4

- Detect the problem
  - Difficult people typically
    - Don't show respect
    - Behave clueless
    - Are overly emotional
    - Make sweeping claims

# Dealing with Difficult People 4 / 4

- Handle the problem
  - Don't engage with difficult people
    - Cf. “Don't feed the troll”
  - Ignore them if possible
  - Boot them, if necessary



# The Volunteering Process (Recap)

#	Volunteer View	Project View
1	Find project	Market project
2	Understand project	Explain project
3	Engage with project	Engage with volunteer
4	Work within project	Work with contributor
5	Lead project	Enable career

# Review / Summary of Session

- Open source projects
- Project roles and careers
- The 3-4-5 model of open communities
  - The three principles of open collaboration
  - The four practices of open communication
  - The five stages of open source volunteering

# Thank you! Questions?

**[dirk.riehle@fau.de](mailto:dirk.riehle@fau.de) – <http://osr.cs.fau.de>**

**[dirk@riehle.org](mailto:dirk@riehle.org) – <http://dirkriehle.com> – [@dirkriehle](#)**

# Credits and License

- Original version
  - © 2010-2019 Dirk Riehle, all rights reserved
- Contributions
  - Christoph Lindenmüller by email of 2011-09-01
  - ...