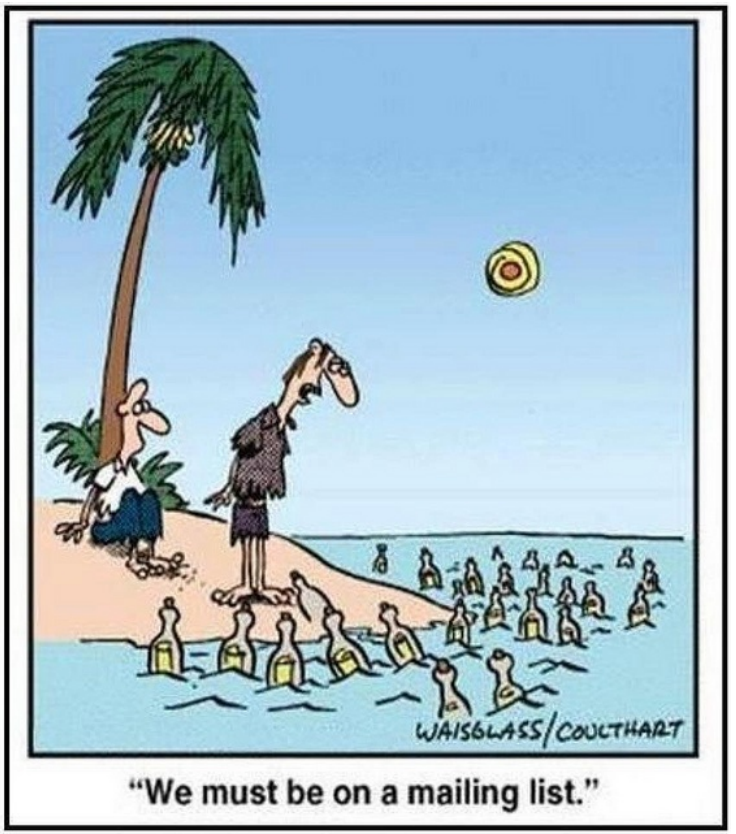
Niels ten Oever
@nielstenoever
niels@digitaldissidents.org

Davide Beraldo
d.beraldo@uva.nl

# Analyzing mailinglists with python




"We must be on a mailing list."

# BigBang



# BigBang?

- Tool for the analysis of collaborative projects

- https://github.com/datactive/bigbang
- https://github.com/nllz/bigbang

- Current focus on:
  - Mailing lists
  - Git repos
  - (a little bit of) ML + Git

# History

- Originally developed by UC Berkeley researcher Sebastian Benthall ~2015
- Picked up by DATACTIVE project last year
- Let’s move it on!

# For what?

- Originally thought for collaborative open source projects (~recursive publics)

- Works great for any organization using mailing lists and/or git (e.g. internet governance bodies)

# Analysis

- Descriptive statistics
- Network analysis
- Text analysis

- Bunch of Jupyter notebooks developed so far

# Libraries

- **Pandas** (dataframe object)
- **NetworkX** (network analysis)
- **Numpy** (data processing)
- **NLTK** (text analysis)

# Core modules

- **bin/collect_mail.py** (collect email data from url or list of urls)
- **bin/collect_git.py** (collect git repos data from names or urls)
- **archive.py** (represent collected email data in a Archive object, computes some basic statistics)
- **graph.py** (generate network graphs and metrics)
- **process.py** (performs some useful data transformation)
- **threads.py** (transform emails into Threads objects)

# Notebooks – Mailinglists

- **analyze senders** --> distribution of top-senders (including smart ways to deal with name/addresses inconsistency)
- **participants over time** --> time series on email senders
- **auditing [single user]** --> follow a single user's contribution over time
- **collaboration robustness** --> various analysis on how relationships between users evolve (duration, reciprocity, etc.)
- **cohort analysis** --> trend over time of new users joining the discussion
- **interaction network** --> generate and visualize graph of interactions in the mailinglists

# Notebooks – Mailinglists 2

- **assortativity study** --> computes degree assortativity on email reply network (whether users tend to interact with likely central users)
- **single word trend** / **multiple word trend** --> trend of a specified word / multiple words in a single / multiple mailinglist
- **special words analysis** --> various interesting ways to identify special words in the mail texts; unique / common words between mailing lists; words introduced by different people in different lists; words introduced by same people in different lists; words that quickly flow from a user to others
- **users top words** --> most recurrent words for specified users
- **who said what words** –> distribution of specified words for specified users
- **threads** --> some analyses about threads in a mailing list: number of threads; distribution of messages per thread; duration of a thread; properties of a single thread

# Notebooks - Git

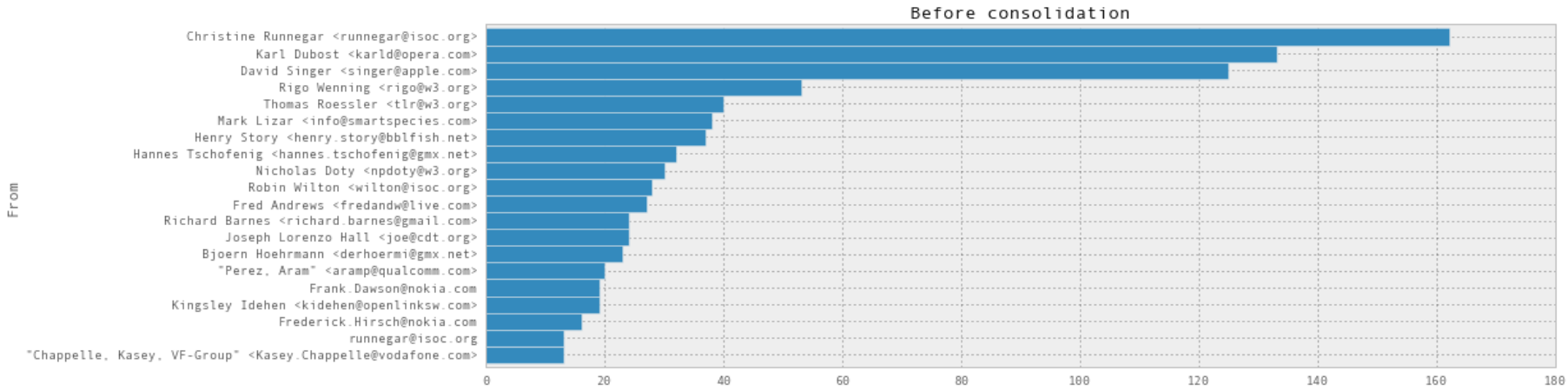
- **committer dominance** --> identify key-contributors to projects
- **git commit network** --> create graph where nodes are commits and edges connect parent and child commits
- **git diffs** --> create bipartite graph where nodes are committers and files, and edges are files edited by committers
- **git interaction graph** --> create graph of collaborations between contributors of a repository
- **multi git repos** --> allows to collect data from multiple repositories, and performs: bi-partite graph with nodes as committers and files and as committers and repositories; projected graph of repositories sharing committers

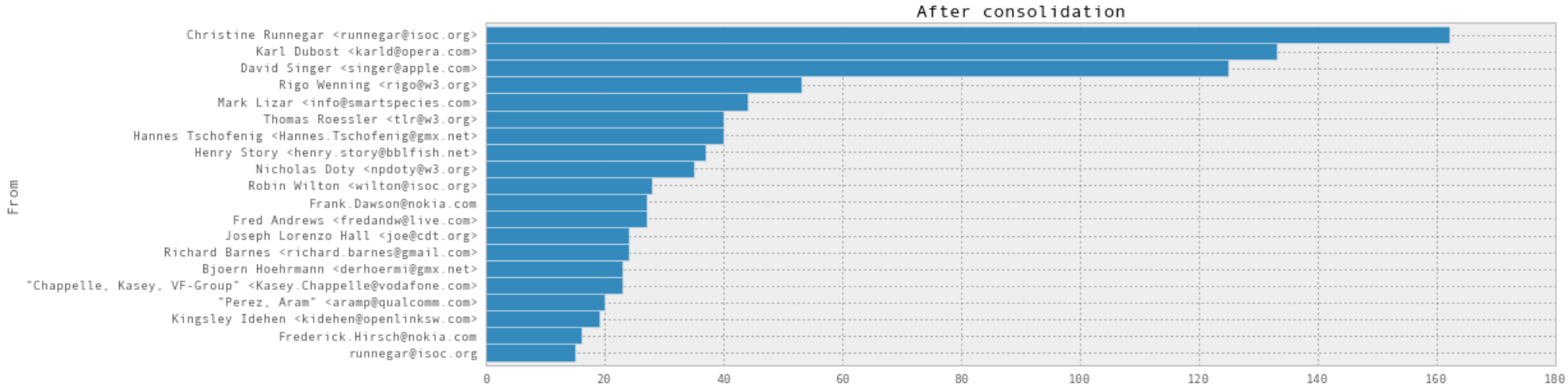
# Notebooks – ML + Git

- **git collection** --> compare trends in commits and emails; identifies top-committers; distribution of contributions by cohorts of committers

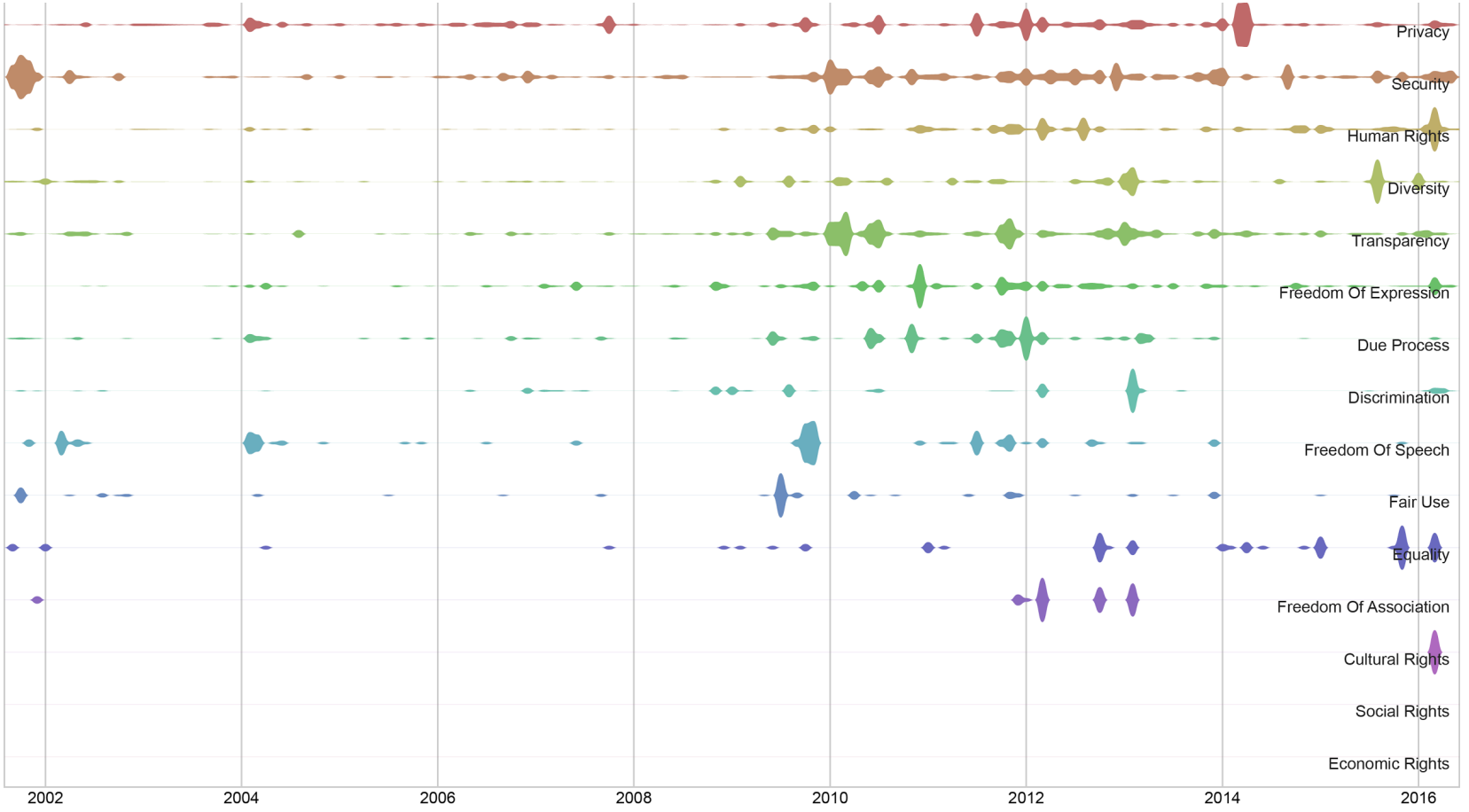
- **workers and talkers** --> compares the activity of contributors to a project in terms of commits and in terms of emails (tests whether those who talk a lot are also those who work or not)

# Analyse senders

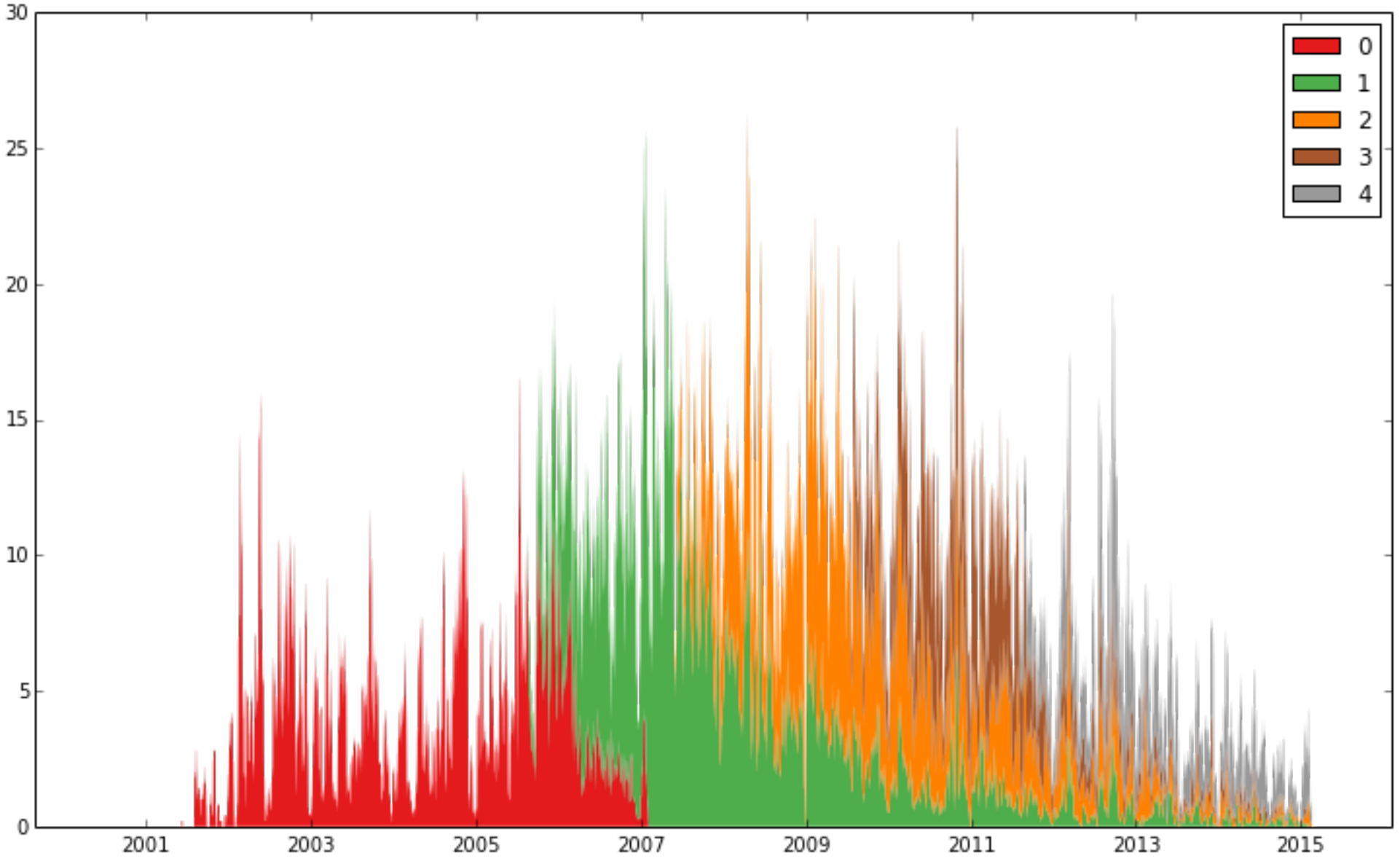


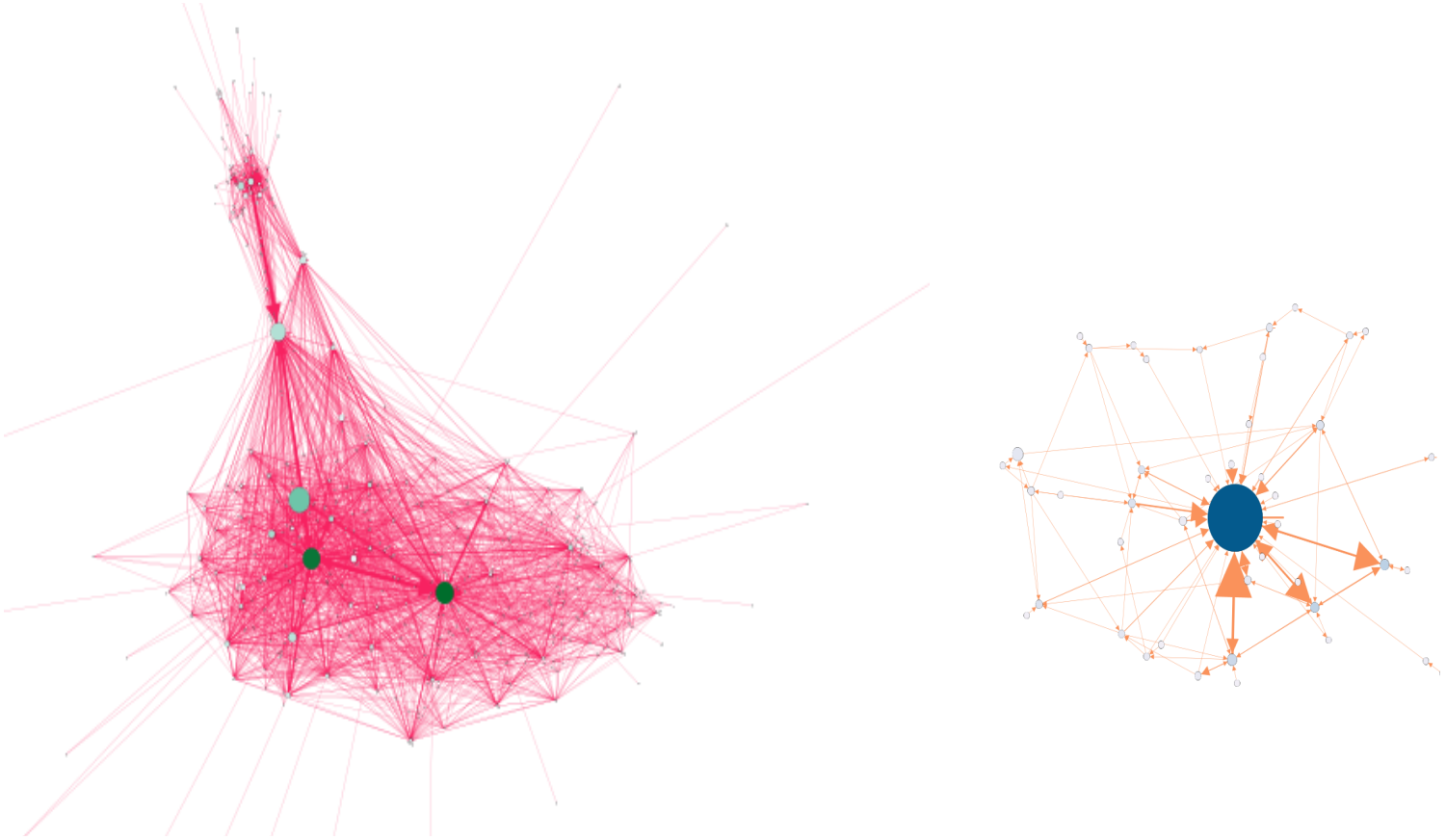
# Multiple words trend



# Cohorts



# Network Graphs



# Open Issues

- Consolidate notebooks into core
- Replace deprecated Pandas.DataFrame methods
- Installation issues on Mac / Windows
- Collect Google Hangout
- Automate NLTK download
- How to deal with Lurkers?
- Implement gender detection
- Inter-Mailinglist network analysis
- (114 and counting...)

# Ethical Concerns

- Public data pulled from publicly accessible mailing list archives (but public != ethical)

- Anonymization of addresses / visualizations

# ICANN example

# Internet Corporation for Assigned Names a Numbers

**ICANN**: Organization & community of stakeholders

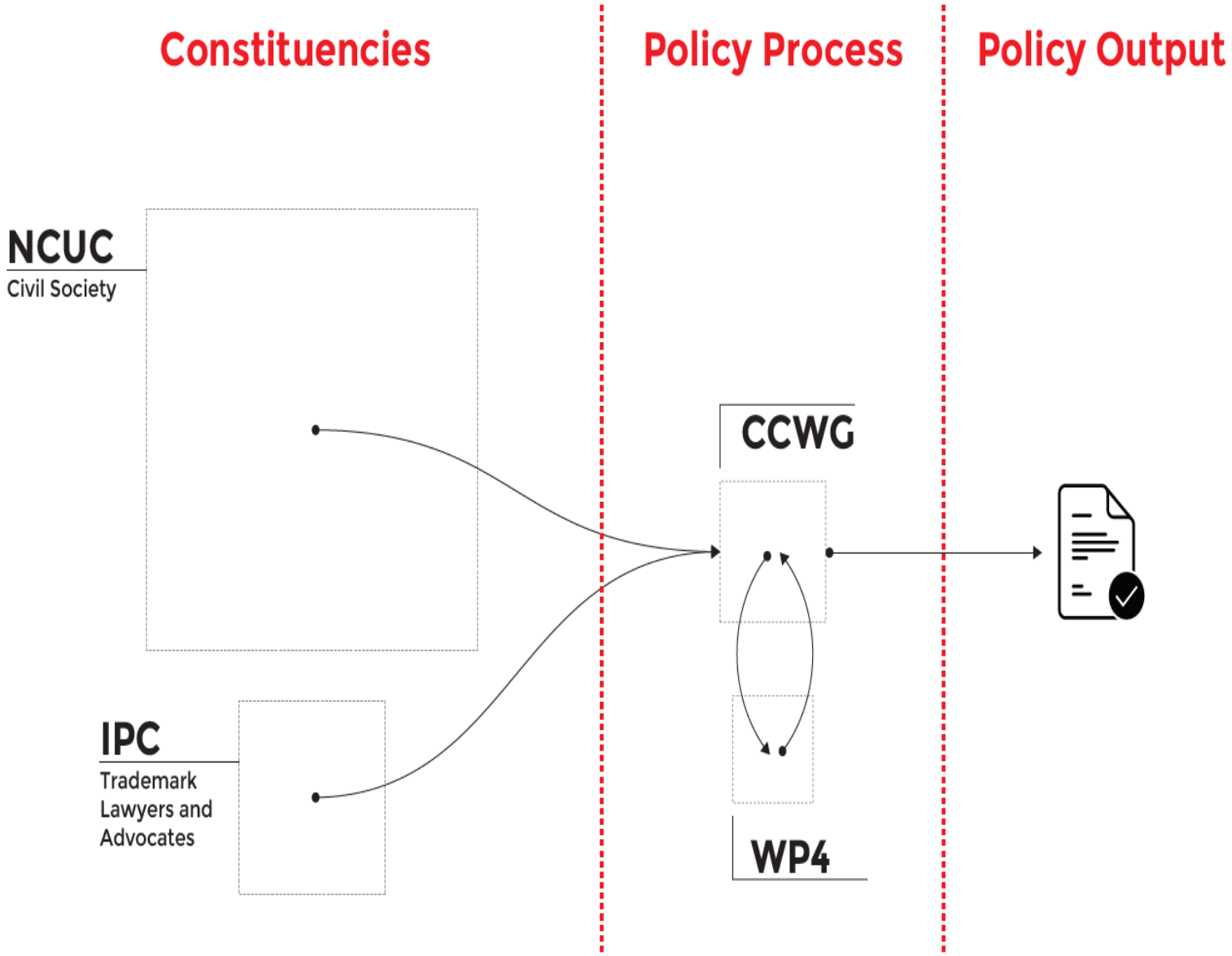
**Organization:** In charge of managing Domain Names System (DNS) and coordination of distribution of IP addresses

**Community**: Develops ICANN policies and procedures

Currently at crucial turning point (transitioning stewardship from US to global multistakeholder community)

**>> How do listserv interactions shape discourses and policy decisions?**

# Process overview

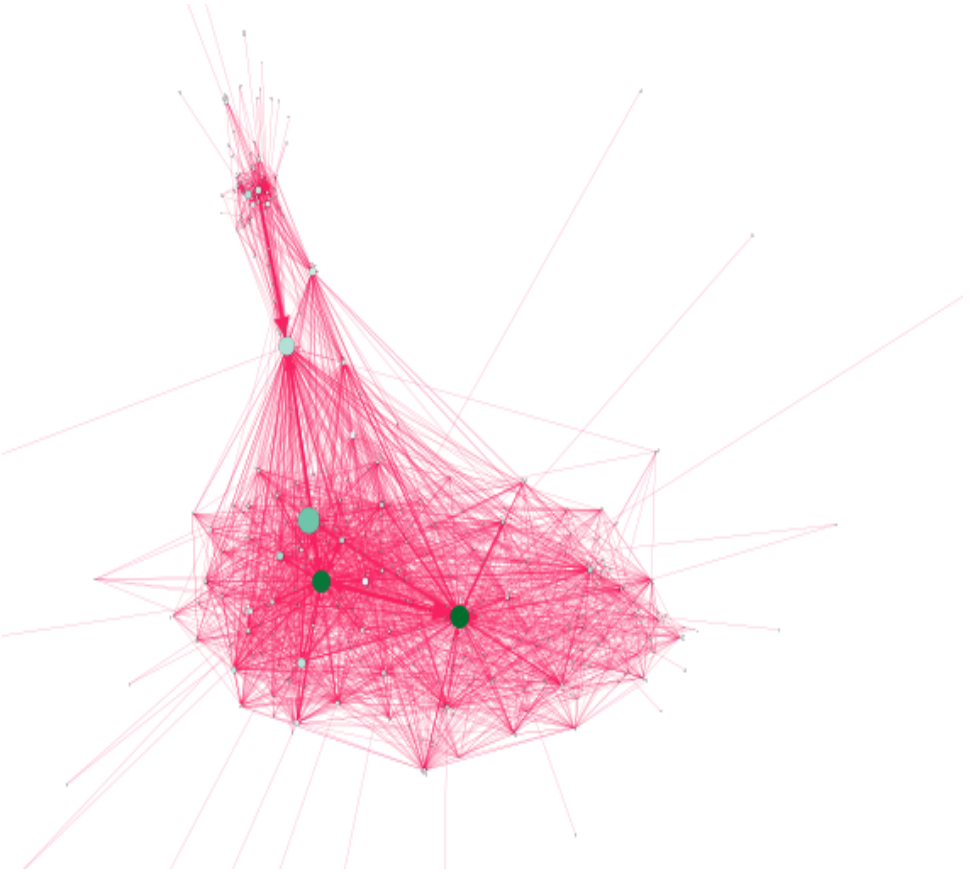


# Relevance (or: why should you care?)

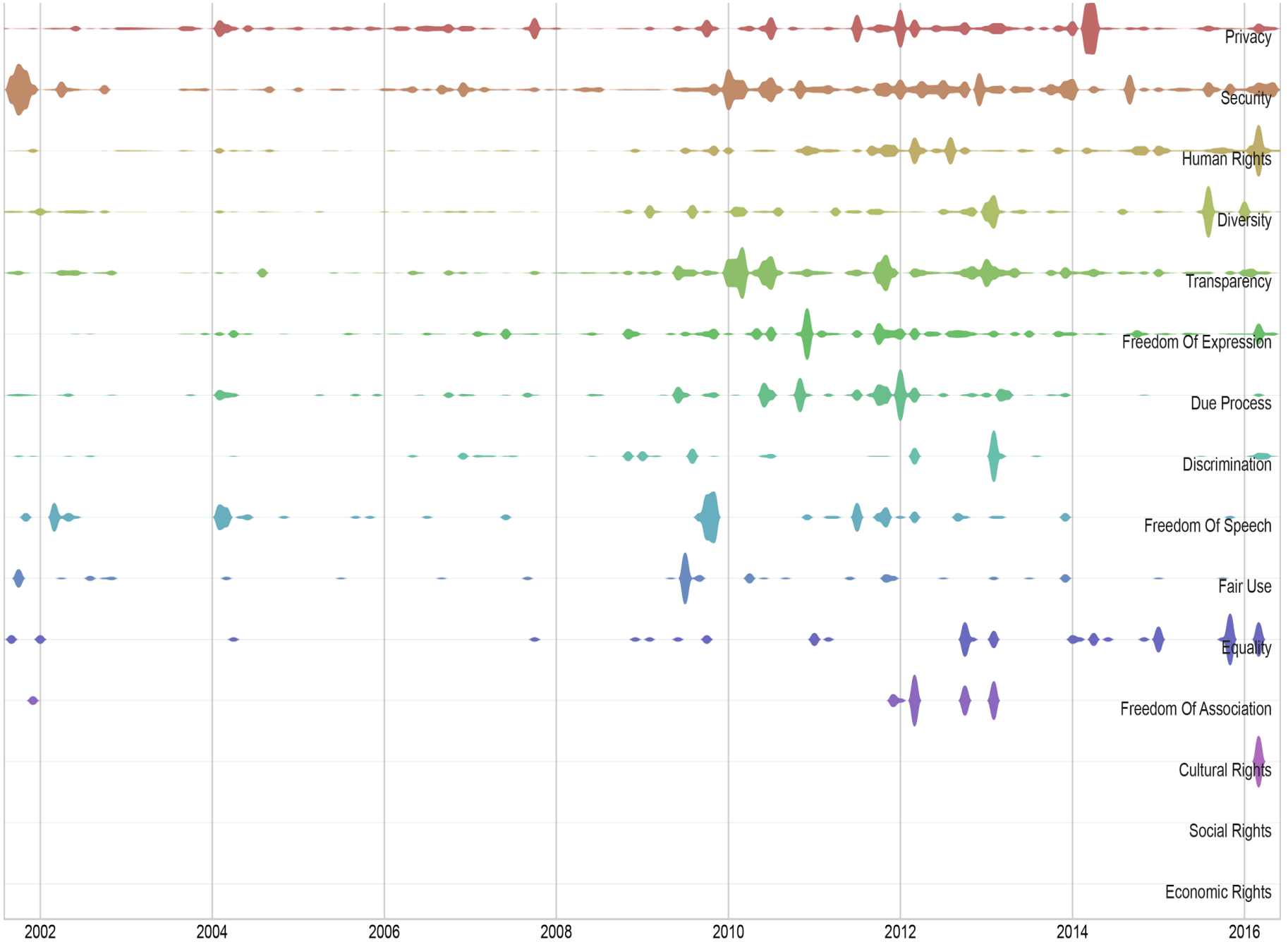
Our research contributes to understanding

- how (policy) discourses evolve in online communities
- network configurations in online communities
- discursive and networking tactics of civil society policy advocacy
- (multistakeholder) institutional design in the field of internet governance

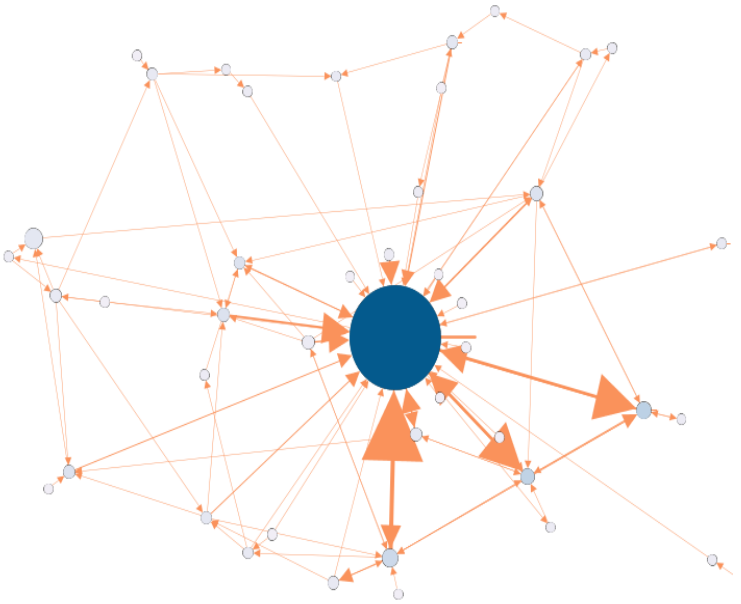
# NCUC: network diagram



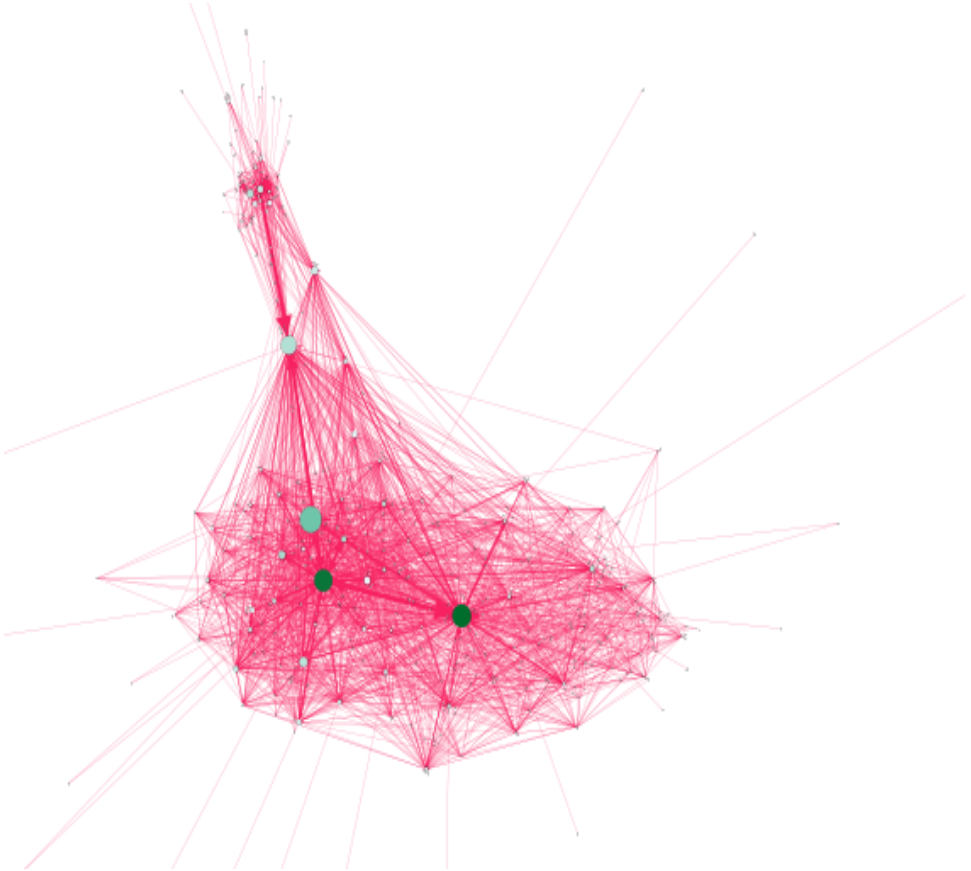
# NCUC: human rights language development

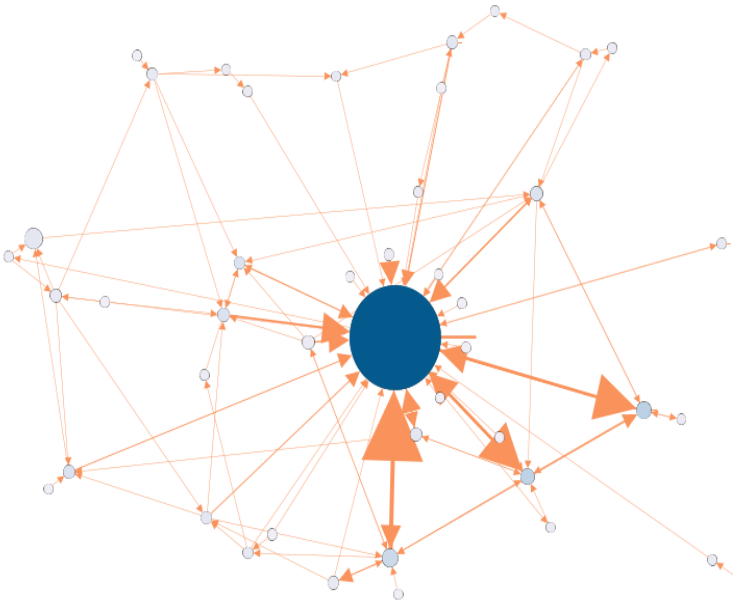


# 3. IPC: network diagram



# Comparing NCUC & IPC





# Policy language (in the 250 most frequently used words)

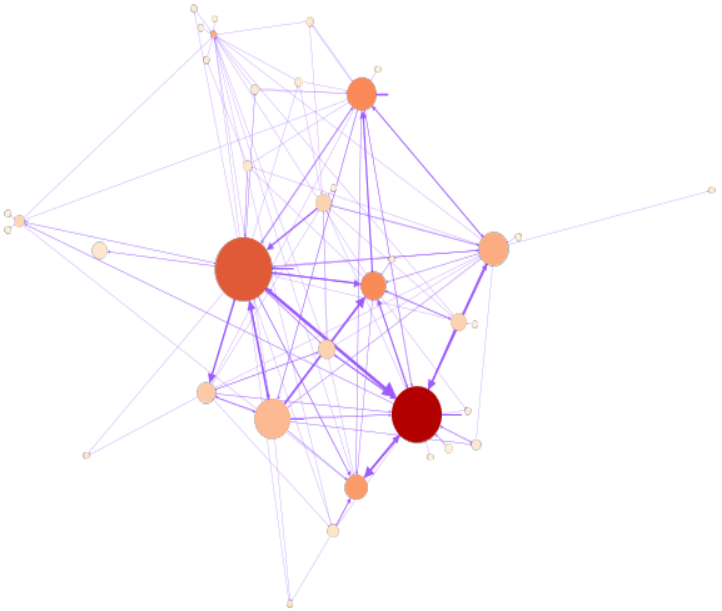

NCUC (civil society)


IPC (trademark interests)

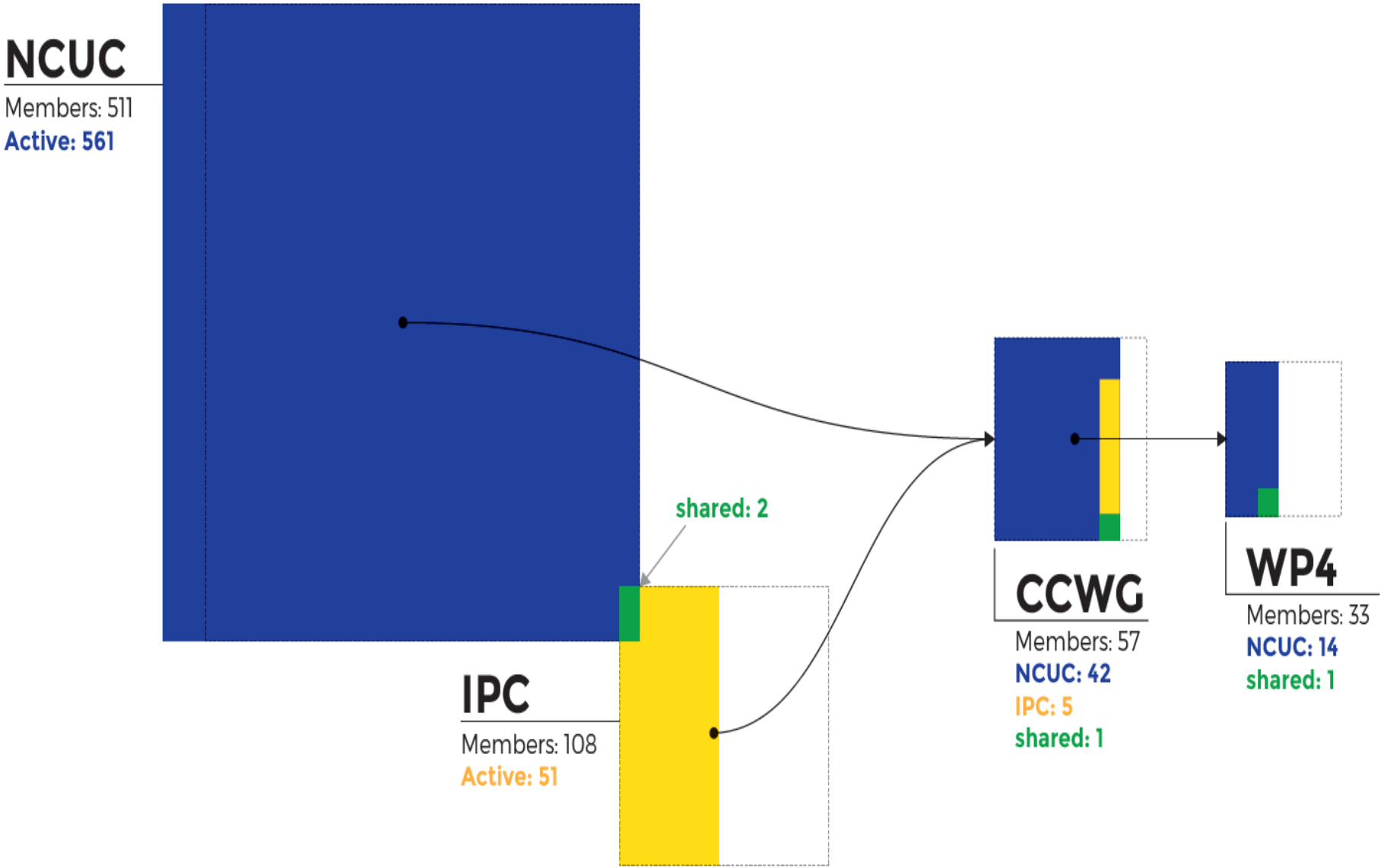
# 4. Comparing CCWG & WP4

**CCWG**
A cross-constituency network dealing with a broad set of issue areas

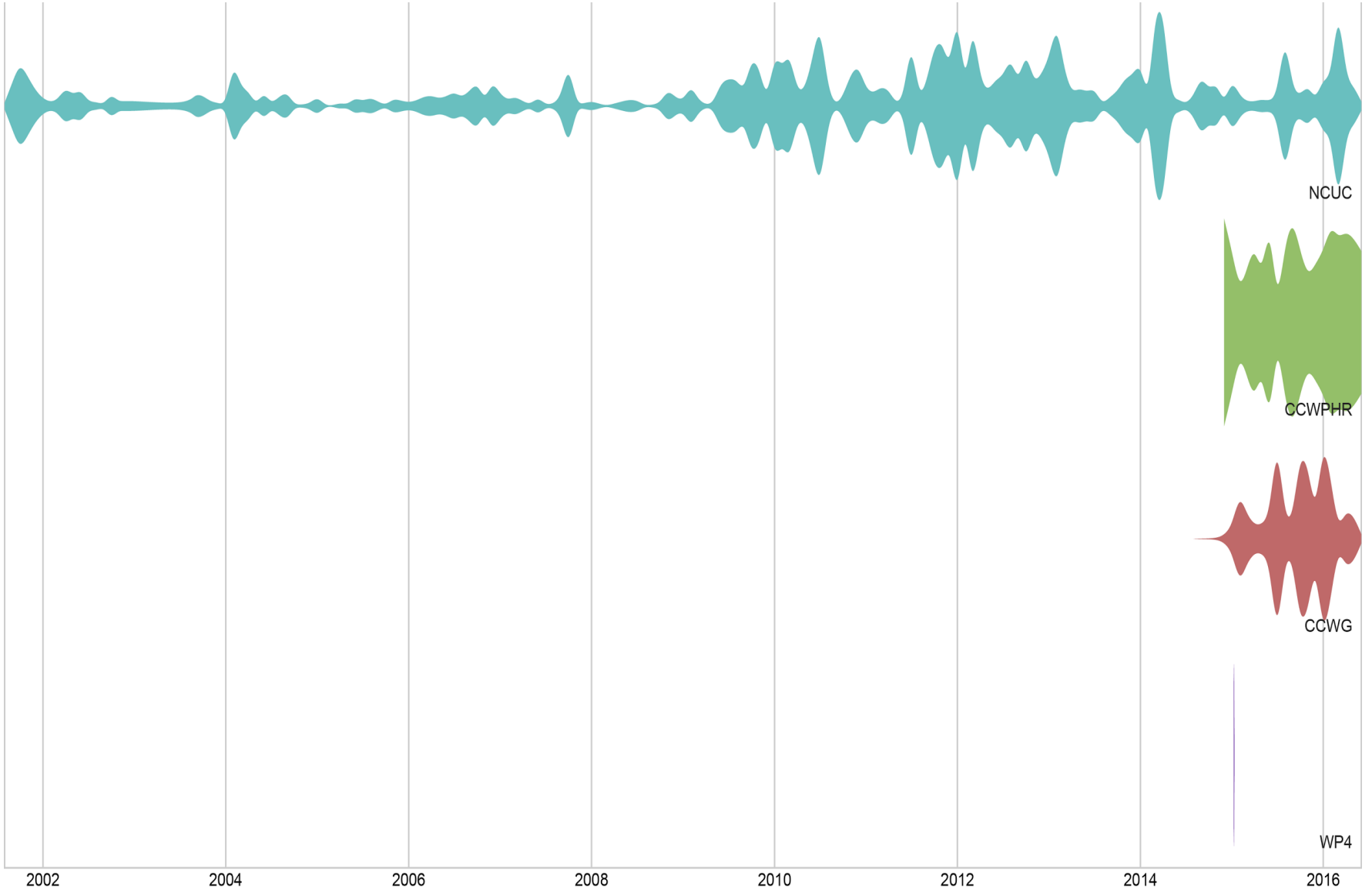
**WP4**
A temporary working group focused exclusively on HRs issues



# Actor Activity and Representation



# Human Rights language through time



# How to

## Installation

We will be working with ipython notebooks. In order to do this you will need to follow these simple steps:

**1. Download Anaconda.**

You can download the installer from here:
Windows: http://repo.continuum.io/archive/Anaconda2-4.0.0-Windows-x86_64.exe
Mac: http://repo.continuum.io/archive/Anaconda2-4.0.0-MacOSX-x86_64.pkg
Linux: http://repo.continuum.io/archive/Anaconda2-4.0.0-Linux-x86_64.sh

**2. Install Anaconda.**

You can simply run the installer, Windows and Mac have a graphical installer, for Linux:
Open a terminal
Go to the download directory (probably: $ cd Downloads)
and then:
$ bash Anaconda2-4.0.0-MacOSX-x86_64.sh

Go for the standard install (so let Anaconda add Anaconda to the bash PATH). Else we'll get into trouble later.

**3. Download and install Git.**

You can download the installer from here:
Windows: https://git-scm.com/download/win
Mac: https://git-scm.com/download/mac
Linux (Ubunu/Debian): open a terminal and type: $ sudo apt-get install git

**4. Clone the BigBang repository.** Open a command prompt (Windows) or a terminal (Mac, Linux) and type:
git clone https://github.com/nllz/bigbang/

**5. Create an Anaconda environment**

conda create -n bigbang python

**6. Install BigBang**

cd bigbang
bash conda-setup.sh (Mac, Linux) conda-setup.sh (Windows)

_-o-__-o-__-o-__-o-_T H A N K S_-o-__-o-__-o-__-o-__-o-__-o-__-o-__-o