

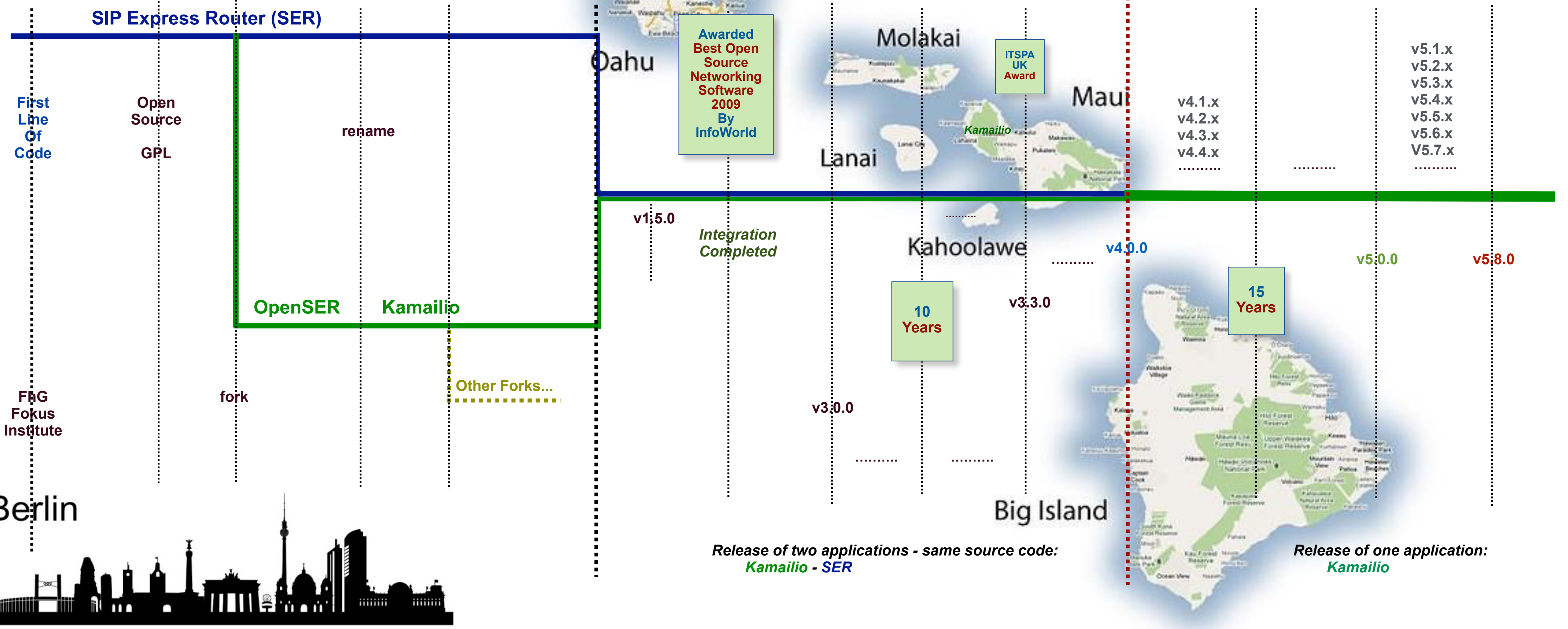
- ▶ Originally from Romania, living in Berlin, Germany
- ▶ Computer science software engineer - Polytechnics University Bucharest (2001)
- ▶ Researcher in RTC at Fraunhofer Fokus Institute, Berlin, Germany (2002-2005)
- ▶ Co-founder, main coordinator and lead developer of Kamailio, an open source SIP Server
- ▶ Professional consultancy for SIP, VoIP, Kamailio and all RTC at asipto.com
- ▶ Involved in open source real time communications since 2002
- ▶ Working with open standard protocols, mainly from IETF, GSMA/3GPP/ITU/ETSI
- ▶ C software developer - mainly VoIP server side infrastructure
- ▶ Co-organizer of Kamailio World Conference, **FOSDEM RTC DevRoom**
- ▶ Speaking and promoting OSS RTC at world wide events





Let's Speak SIP = E Kama'ilio SIP

Sep 2001 2002 Jun 2005 Jul 2008 Aug 2008 Nov 2008 Oct 2009 Jan 2010 Sep 2011 Jun 2012 Mar 2013 Sep 2016 Mar 2017 March 2024

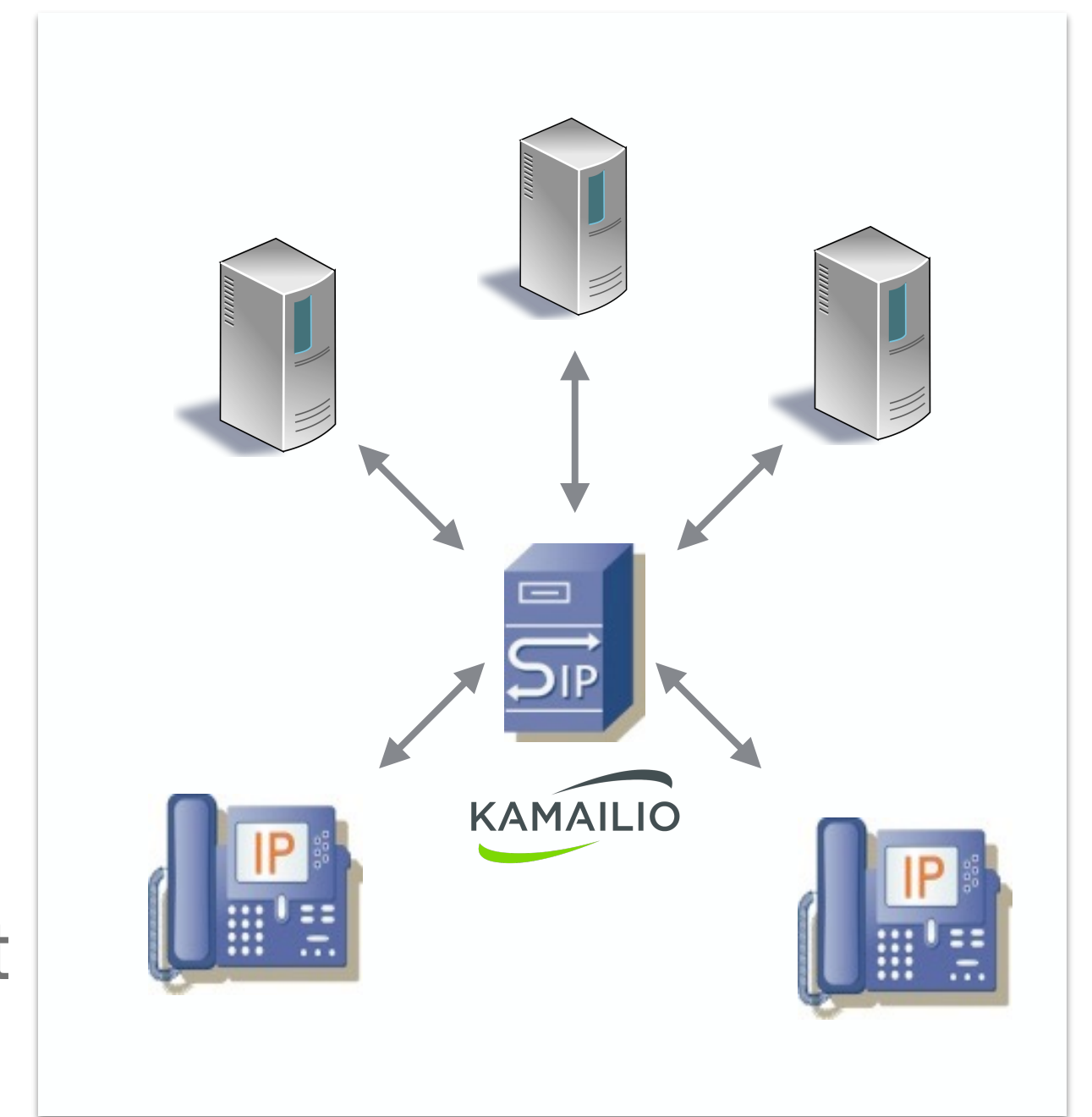


KAMAILIO SIP SERVER - SIGNALING



<https://www.kamailio.org>

- * Open Source SIP (IETF RFC3261) Signaling Server implementation, developed since 2001
- * Can be used for VoIP (Voice, Video, VoNR/VoLTE/IMS, SIP-I/SIP-T), Instant Messaging, SMS, Presence
- * Diameter, SQL and NoSQL backends
- * load balancing, least cost routing, security gateways
- * Designed for modularity, flexibility and scalability
 - * used by large telecoms, mobile operators and OTT services world wide
 - * thousands of call setups per second
 - * hundred thousands of connected phones per instance
- * IPv6/IPv4 - UDP/TCP/TLS/SCTP/WebSocket - asynchronous routing
- * **Classic SIP - WebRTC gateway using Kamailio + RTPEngine**
- * Embedded interpreters: Lua, Python, JavaScript, Ruby, Squirrel, Perl, .Net
- * About 250 modules (extensions)



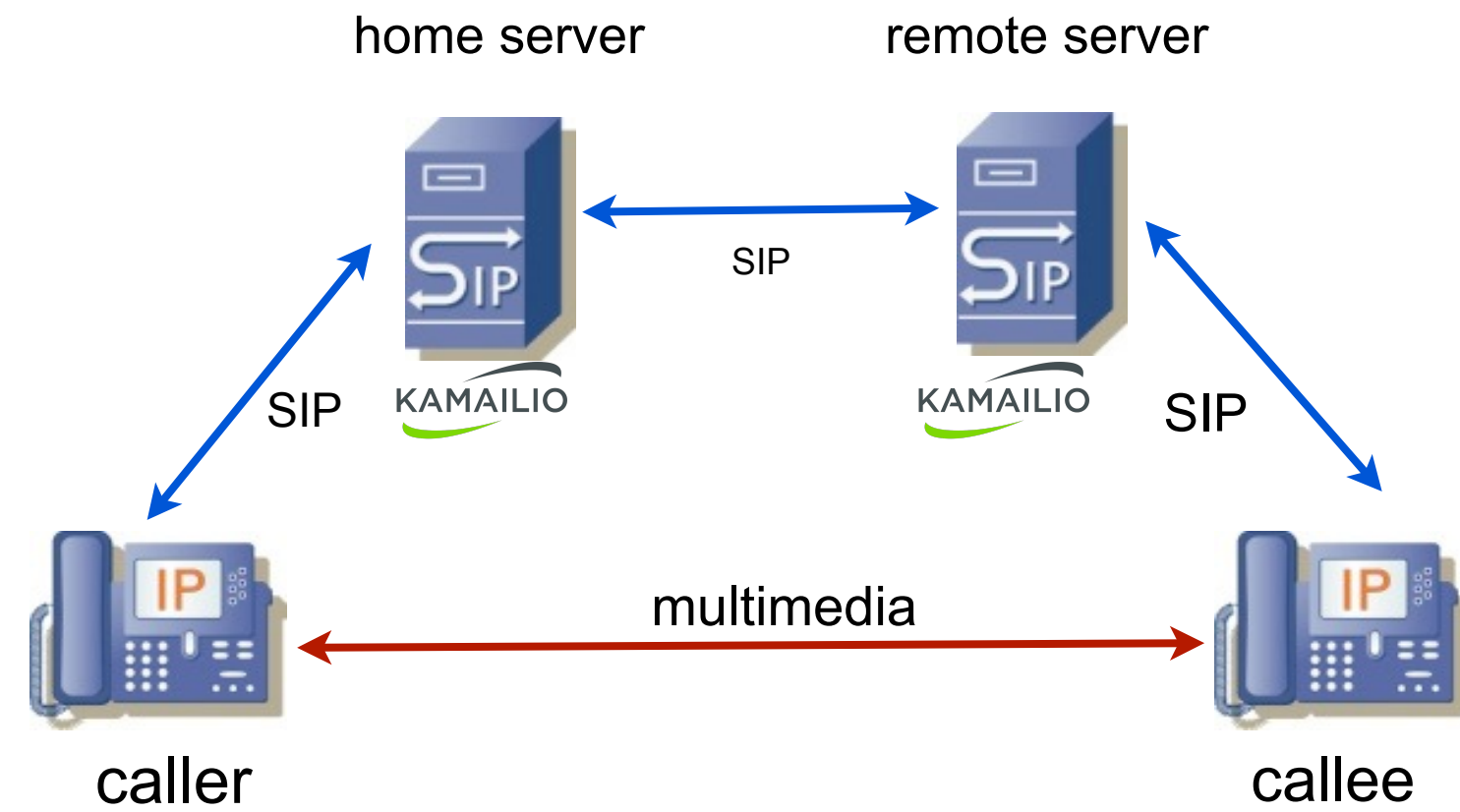
KAMAILIO - SIP ROUTING TOOLKIT - BUILDING BLOCKS



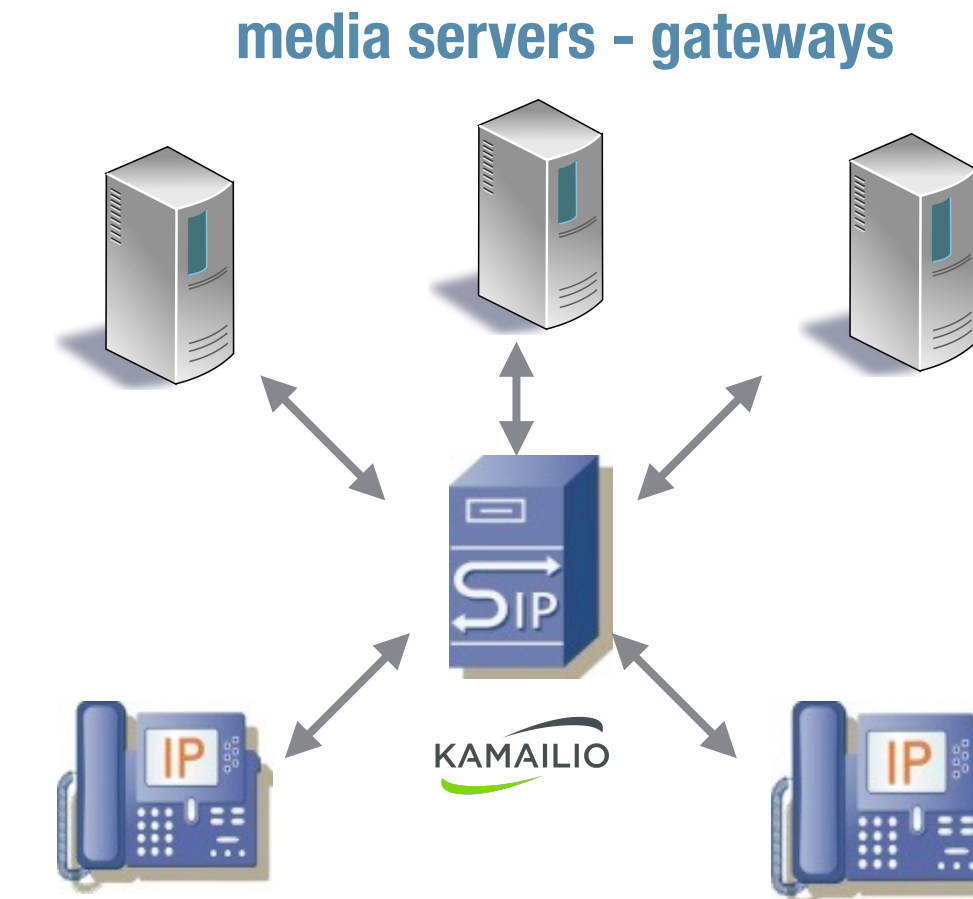
- * Authentication
- * Authorization
- * Accounting
- * Registration
- * Location
- * Least cost routing
- * Load balancing
- * Encryption
- * Diameter
- * IMS Extensions
- * **HTTP (Rest) API client-server**
- * **JSON, XML, JWT**
- * **WebSocket, WebRTC (with RTPEngine)**
- * ...



COMMON USES CASES



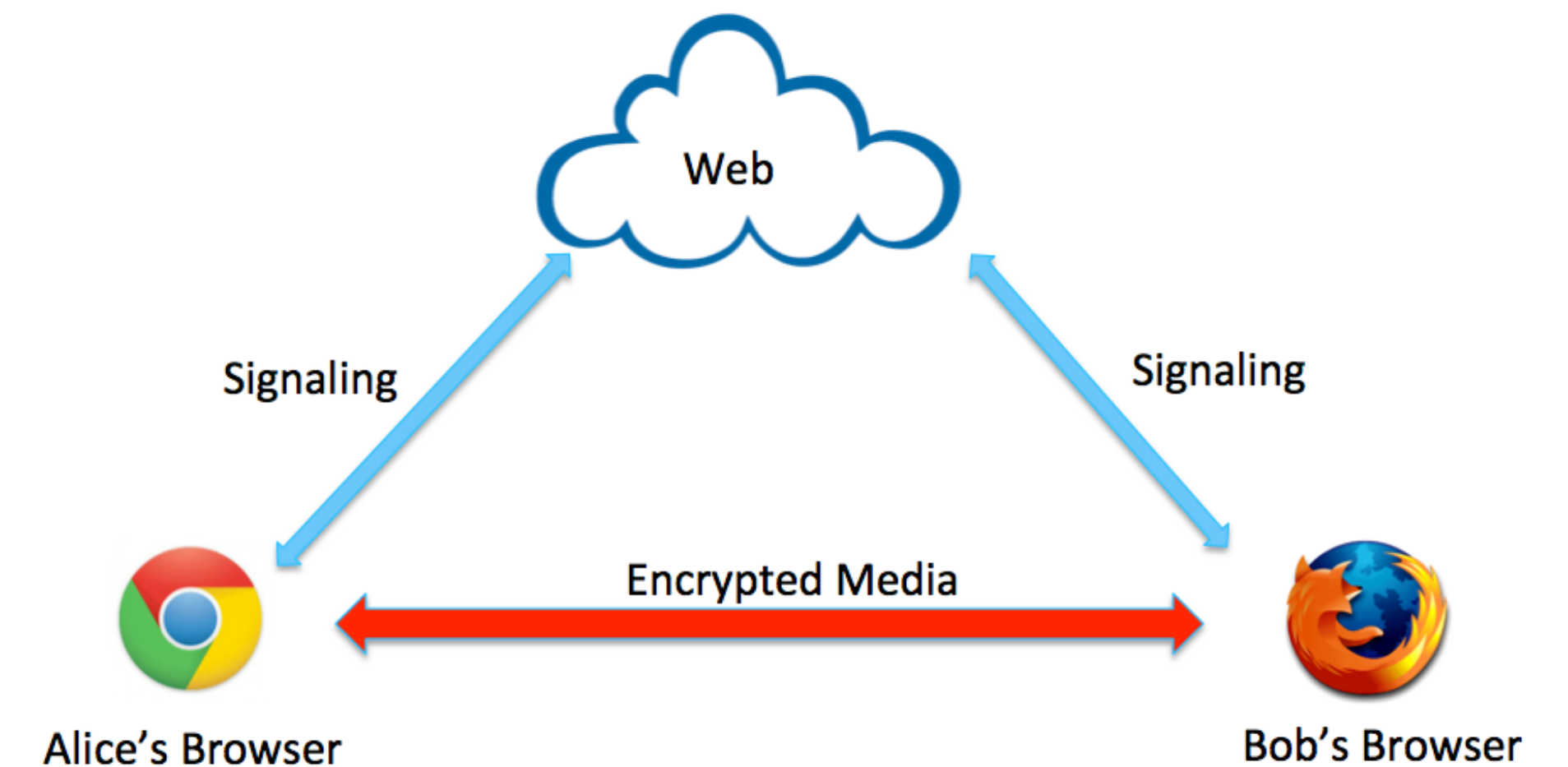
- * authentication, registration and user location
- * voice, video, instant messaging and presence
- * NAT traversal, RTP relaying, web rtc
- * SIP security firewall
 - * DDoS mitigation, anti-fraud
- * integration with social networking



- * load balancer
- * least cost routing
- * transport layer gateway
- * topology hiding
- * carriers interconnect

KAMAILIO IN THE WEBRTC WORLD

- * Support since 2012
 - * the first open source SIP server implementing it
- * Kamailio - signalling handling
 - * websocket (+ xhttp, tls) module
 - * rtpengine (+ nathelper) module
- * RTPEngine - media handling
 - * ICE, encryption-decryption, transcoding
- * JavaScript SIP libraries
 - * JsSIP, SIP.js, SIPML5, ...
- * Use cases
 - * Browser to browser calling
 - * WebRTC to classic SIP/VoIP gateway



KAMAILIO IN THE WEBRTC WORLD

- * Kamailio for SIP signalling routing
- * RTPEngine for RTP-SRTP encryption-decryption
 - * <https://github.com/sipwise/rtpengine>

<http://www.kamailio.org/docs/modules/stable/modules/rtpengine.html>

<http://www.kamailio.org/docs/modules/stable/modules/websocket.html>

<https://gist.github.com/jesusprubio/4066845>

<https://github.com/caruidiaz/kamailio-ws>

<https://github.com/havfo/WEBRTC-to-SIP>

SECSIPIDX PROJECT - STIR/SHAKEN

<https://github.com/asipto/secsipidx>

Components:

- **secsipid**: Go library - common functions
- **csecsipid**: C library - wrapper code to build dynamic or static library and .h include files
- **secsipidx**: `main.go` - CLI tool and HTTP API server for checking or building SIP identity

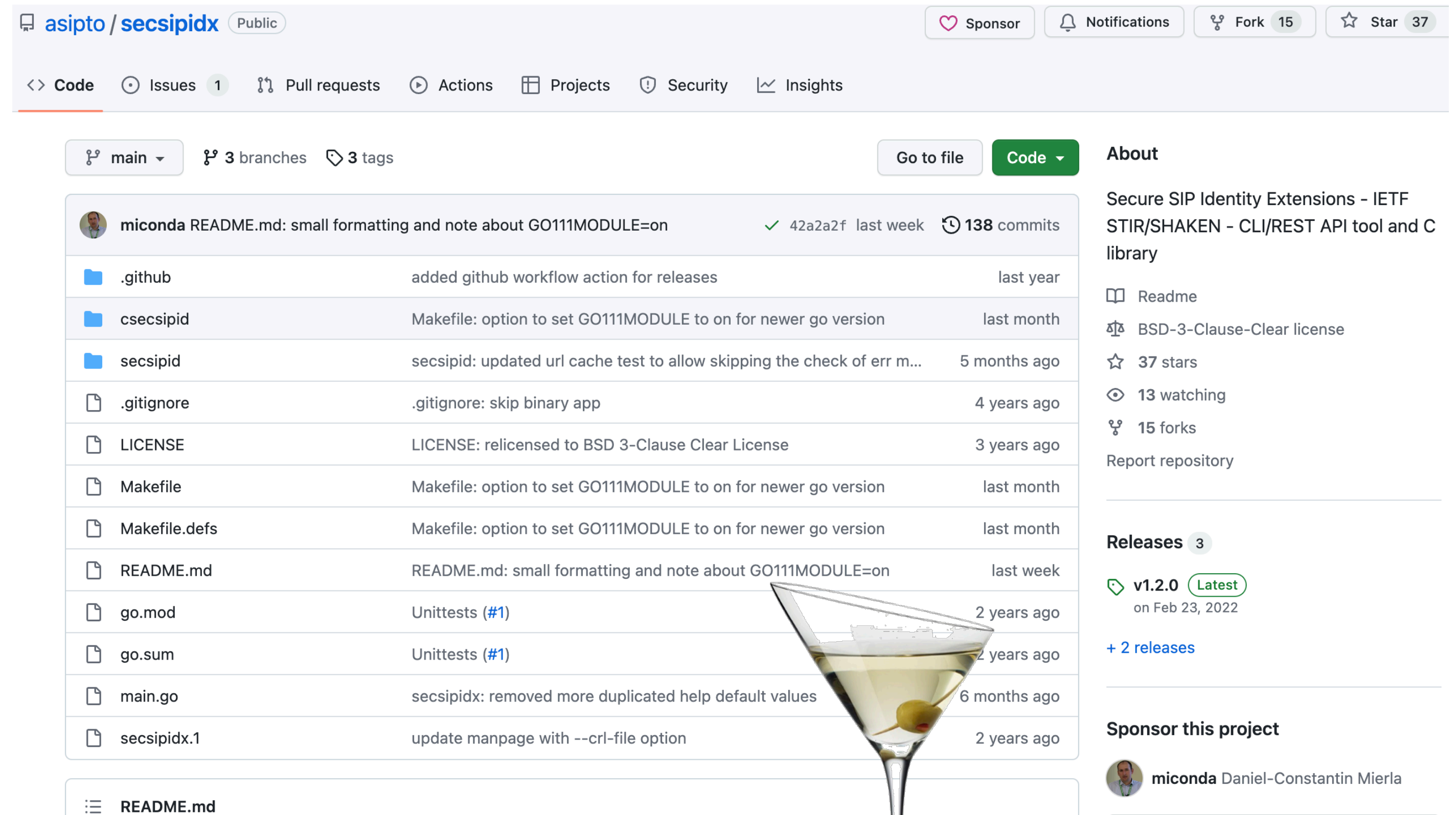
STIR (Secure Telephony Identity Revisited)

- a series of IETF RFCs: RFC8224, 8225, 8226
- <https://tools.ietf.org/html/rfc8224>

SHAKEN (Secure Handling of Asserted information using toKENs)

- RFC8588 - <https://tools.ietf.org/html/rfc8588>

They defines how telephone service providers should work together to ensure calling numbers have not been spoofed.



asipto / secsipidx Public

Sponsor Notifications Fork 15 Star 37

<> Code Issues 1 Pull requests Actions Projects Security Insights

main 3 branches 3 tags Go to file Code

About
Secure SIP Identity Extensions - IETF STIR/SHAKEN - CLI/REST API tool and C library
Readme
BSD-3-Clause-Clear license
37 stars
13 watching
15 forks
Report repository

Releases 3
v1.2.0 Latest on Feb 23, 2022
+ 2 releases


Sponsor this project
miconda Daniel-Constantin Mierla

File/Folder	Description	Last Commit
miconda	README.md: small formatting and note about GO111MODULE=on	42a2a2f last week 138 commits
.github	added github workflow action for releases	last year
csecsipid	Makefile: option to set GO111MODULE to on for newer go version	last month
secsipid	secsipid: updated url cache test to allow skipping the check of err m...	5 months ago
.gitignore	.gitignore: skip binary app	4 years ago
LICENSE	LICENSE: relicensed to BSD 3-Clause Clear License	3 years ago
Makefile	Makefile: option to set GO111MODULE to on for newer go version	last month
Makefile.defs	Makefile: option to set GO111MODULE to on for newer go version	last month
README.md	README.md: small formatting and note about GO111MODULE=on	last week
go.mod	Unittests (#1)	2 years ago
go.sum	Unittests (#1)	2 years ago
main.go	secsipidx: removed more duplicated help default values	6 months ago
secsipidx.1	update manpage with --crl-file option	2 years ago

☰ README.md


SIPEXER - SIP CLI TOOL WITH WEBSOCKET SUPPORT












<https://github.com/miconda/sipexer>

 **miconda / sipexer** Public Sponsor Notifications Fork 20 Star 178

[Code](#) [Issues 2](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

[main](#) 1 branch 7 tags [Go to file](#) [Code](#)






 **miconda** README.md: small update for \$uuid 9d721be last week 🕒 231 commits

 .github	Create FUNDING.yml	last year
 misc/img	added gif with executing sipexer to send SIP OPTIONS request	last year
 sgsip	sgsip.go: building messages with sdp late offer	8 months ago
 test	added file with test code snippets	2 years ago
 .gitignore	.gitignore: skip vendor folder	last year
 LICENSE	Initial commit	2 years ago
 README.md	README.md: small update for \$uuid	last week
 go.mod	go.mod updates	last year
 go.sum	added x/websocket package dependency	2 years ago
 selog.go	sipexer: option to enable color output	2 years ago
 sipexer.go	sipexer: adjust cli options printing for value and type suffixes	4 months ago


[README.md](#)

About
Modern and flexible SIP/VoIP cli tool

[go](#) [golang](#) [sip](#) [websocket](#)
[webrtc](#) [telephony](#) [voip](#)

 [Readme](#)
 [GPL-3.0 license](#)
 **178 stars**
 **11 watching**
 **20 forks**
[Report repository](#)

Releases 7

 **v1.0.3** Latest
on Jul 12, 2022

[+ 6 releases](#)

Sponsor this project

SIPEXER - SIP CLI TOOL WITH WEBSOCKET SUPPORT

* Among features:

- * send OPTIONS request (quick SIP ping to check if server is alive)
 - * do registration and un-registration with customized expires value and contact URI
 - * authentication with plain or HA1 passwords
 - * set custom SIP headers
 - * template system for building SIP requests
 - * fields in the templates can be set via command line parameters or a JSON file
 - * variables for setting field values (e.g., random number, data, time, environment variables, uuid, random string, ...)
 - * simulate SIP calls at signaling layer (INVITE-wait-BYE)
 - * option for late-offer SDP
 - * respond to requests coming during SIP calls (e.g., OPTIONS keepalives)
 - * send instant messages with SIP MESSAGE requests
 - * color output mode for easier troubleshooting
 - * support for many transport layers: IPv4 and IPv6, UDP, TCP, TLS and **WebSocket (for WebRTC)**
 - * send SIP requests of any type (e.g., INFO, SUBSCRIBE, NOTIFY, ...)
-
- * **wsctl** - <https://github.com/miconda/wsctl>

OPEN SOURCE

COSTS AND BENEFITS

(OBVIOUS OR HIDDEN)

PROGRAMMING LANGUAGE AND DEPENDENCIES

- * C programming language
 - * very good performance
 - * very good portability*
 - * across Unix/Linux/BSD systems
 - * constrained by dependencies
 - * hard to attract new contributors
 - * syntax allows complex statements
 - * hard to understand and maintain
- * using external libraries
 - * rapid development
 - * reuse of knowledge and resources
 - * one can't implement everything
 - * can incur significant overhead
 - * complexity and break of API: libssl
 - * no longer properly maintained: libev



- * use of standards
 - * compatibility and interworking
 - * standardisation groups collide
 - * xml namespace
 - * http2 - http11+upgrade vs. direct http2
 - * self-brewed solutions became a pain over the time



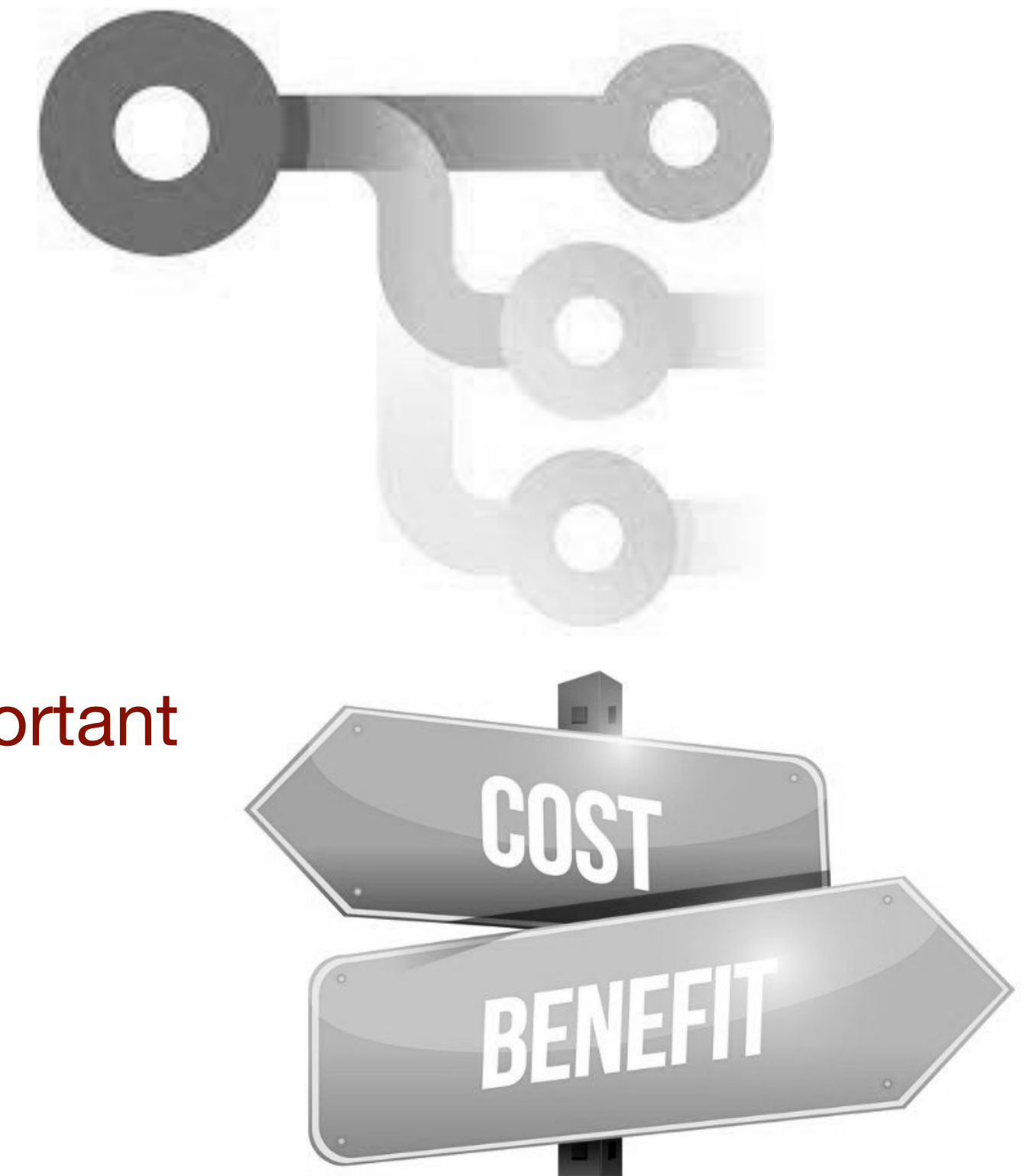
CONTRIBUTIONS AND CONTRIBUTORS

- * the heart and spirit of open source
- * growing the set of feature and user base
- * good impact on quality of the project
- * mix of styles
 - * design and architecture of the components
 - * coding styles - names for functions and variables
- * different levels of experience
 - * hacks and workarounds
- * claims of full ownership even after small contributions
- * friendly relations might not last forever
- * contributions guidelines
 - * coherent format and descriptive commit message
 - * formatting the code with same tool
 - * small commits
 - * easier to understand
 - * modularity
 - * do not impact everything



PROJECT FORKING

- * soft forks - aka cloning git repository of a project to the personal account
 - * how pull requests can be done by external contributors
 - * can foster development of new components and features
 - * bug fixes and testing
 - * can take some resources to track and merge back
- * hard forks - cloning, renaming and diverging
 - * they are mostly noise
 - * every little different thing is made up to look as something important
 - * they diverge anyhow, not worthing anything after a while
 - * better to ignore, follow the need of your project and community



COMMUNITY

- * where all the fun is, and the pain
- * learning a lot from others on different topics
- * source of good references
- * meeting people when travelling and making friends

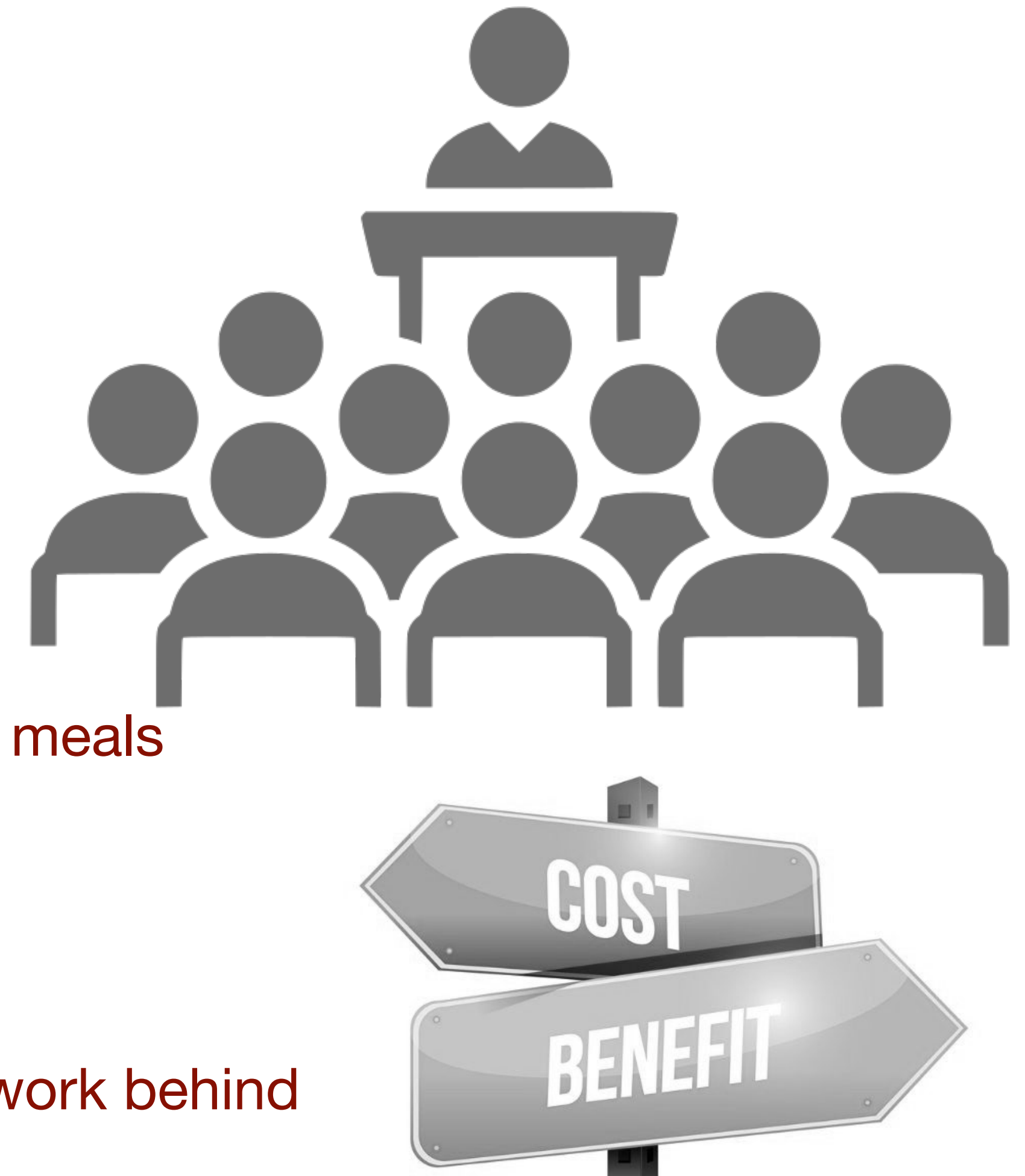
- * aggressive and sometime abusive attitude
- * blaming the developers, even everything is free
 - * treating as being their employees
- * not following guidelines
- * inexact description of reports
- * not keeping the promises
 - * e.g., help me and I will make documentation for it

- * keep focus on fair users and contributors, and ignore the intruders



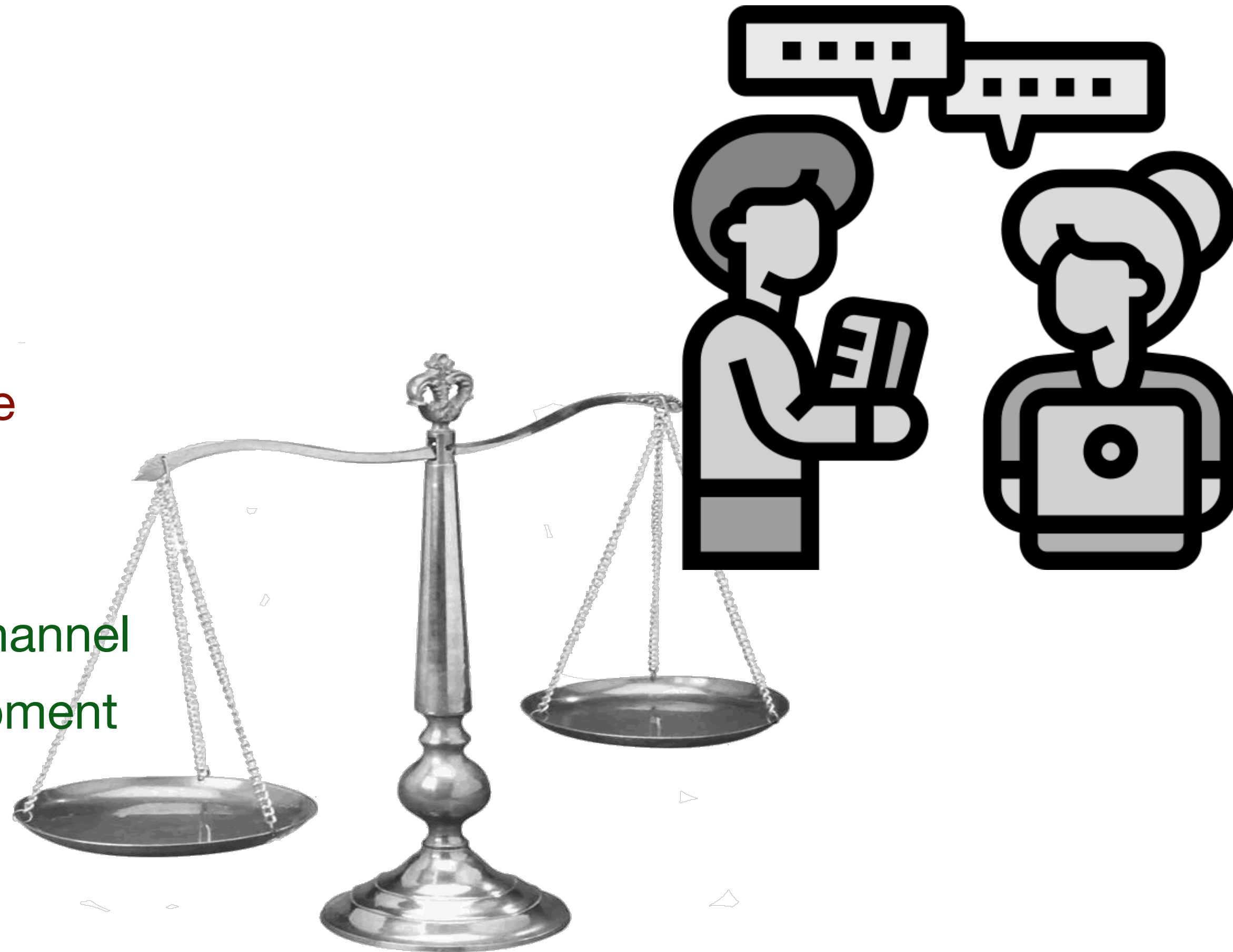
EVENTS

- * very useful community meetings and conferences
 - * sharing experiences
- * development workshops
 - * boost collaboration and speed up evolution
- * free events can be disappointing
 - * many register and do not show up
 - * charge a fee, which can be returned on site as a gift, drinks, meals
- * others feel entitled of ownership and special benefits
 - * organisers should accept their instructions and rules
 - * participate for free or at lower registration fee
 - * because all runs smooth, they do not think of the risks and work behind
- * make clear the rules of participation, the organiser and the scope



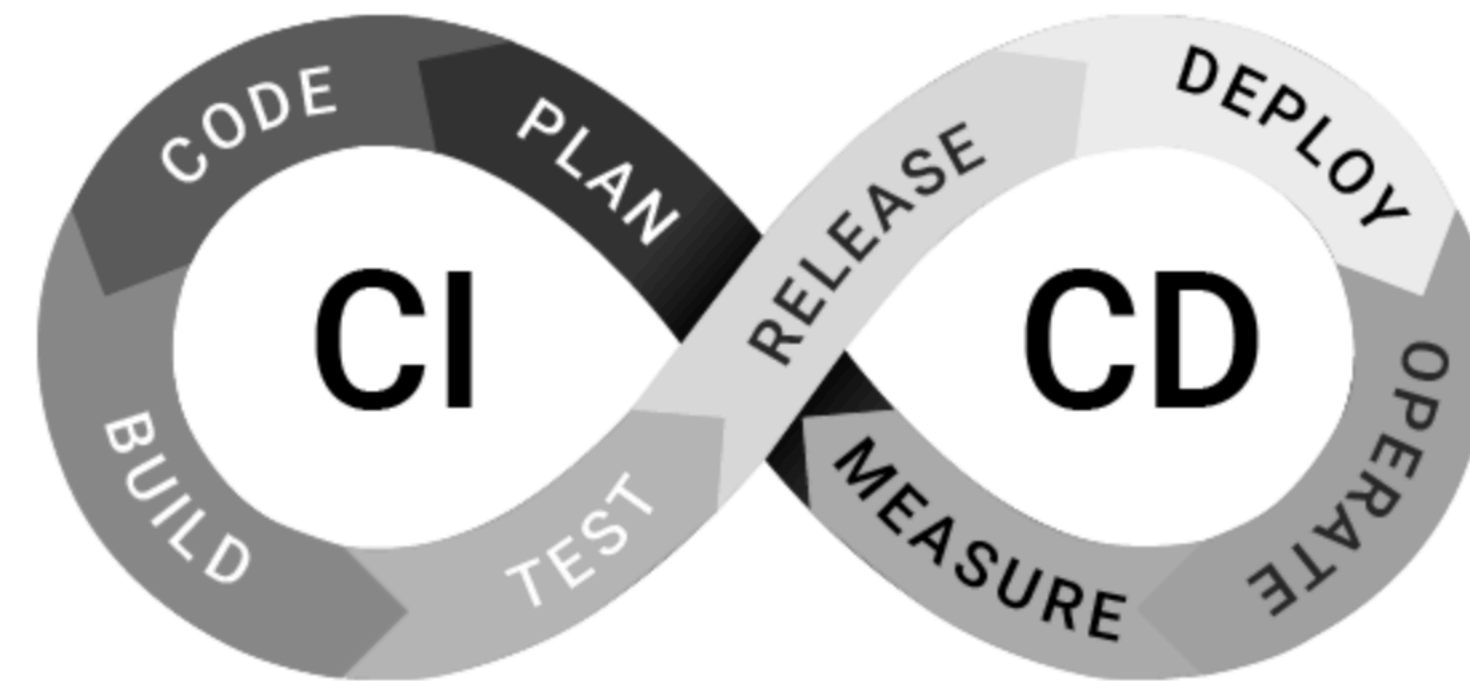
COMMUNICATIONS CHANNELS

- * the ways the community interacts and can collaborate
- * many options for the same scope
 - * mailing lists, forums, chat rooms
 - * trackers or support portals
- * can distract activity
- * hard to track on topics and purpose
- * maintenance costs
- * clearly define the scope for each channel
 - * bug reports and tracking development
 - * how-to use discussions
 - * business relations



TESTING & CI/CD

- * Ensure minimum coherence and acceptance testing
 - * Github actions are very useful
 - * Docker, Jenkins
 - * Detect simple or involuntary mistakes
 - * Lift burden from developers
 - * Community can play a key role
-
- * Installation, configuration and maintenance costs
 - * Ownership and privacy issues
-
- * Best testing is in production

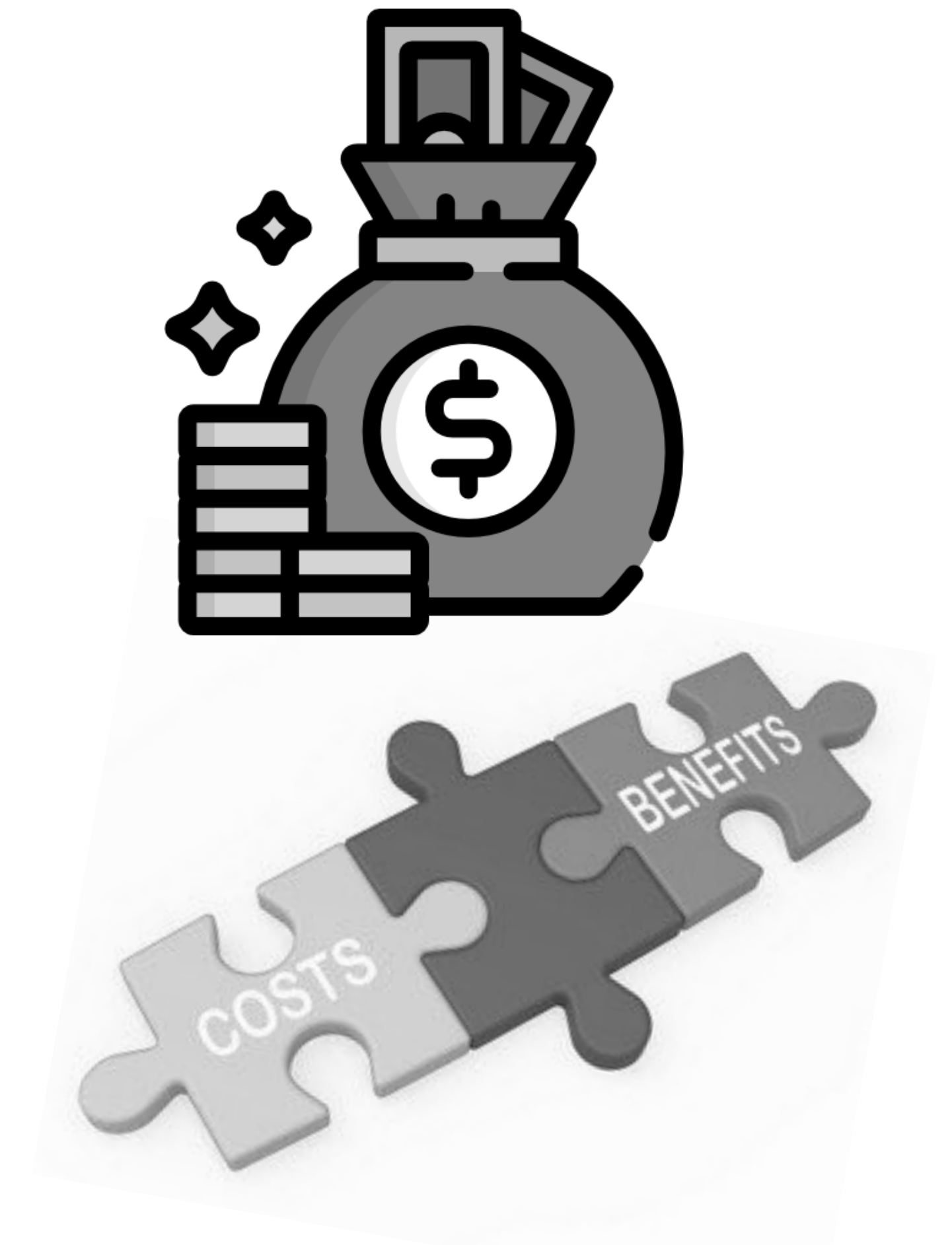


REVENUE - DONATIONS

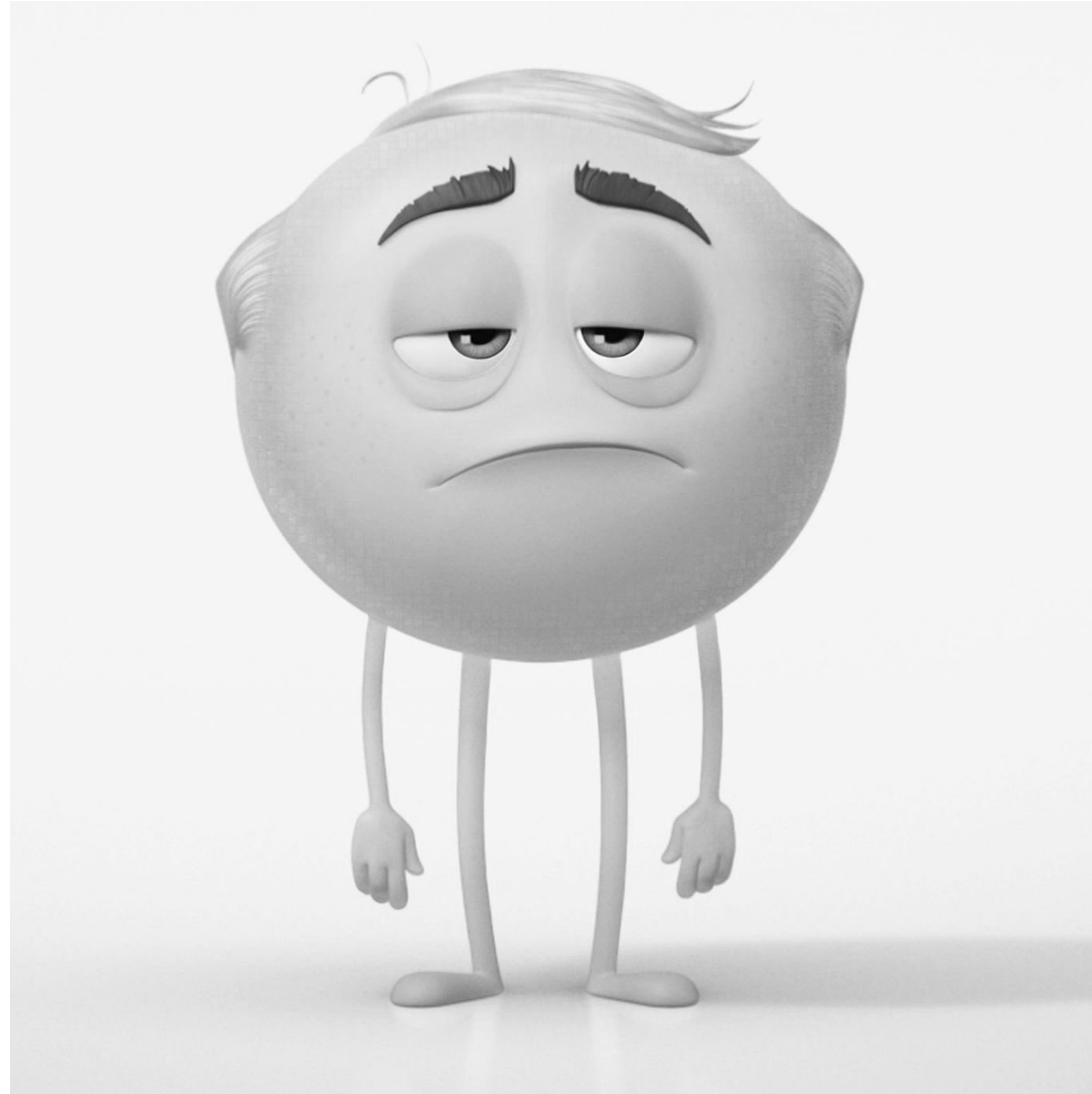
- * open source is a channel for marketing
 - * sometimes overrated
- * free drinks, dinners and some donations
- * boost of career opportunities
- * several directions: development, support, training, SaaS, CPaaS

- * OSS makes it easy to get distracted and lost in irrelevant matters
 - * negative impact on family and social life
- * do not bet the life on donations
- * support revenue might not be as much as expected
 - * or not desired
- * sponsorships can put pressure on developers
- * fairness of others might not be as developers expect

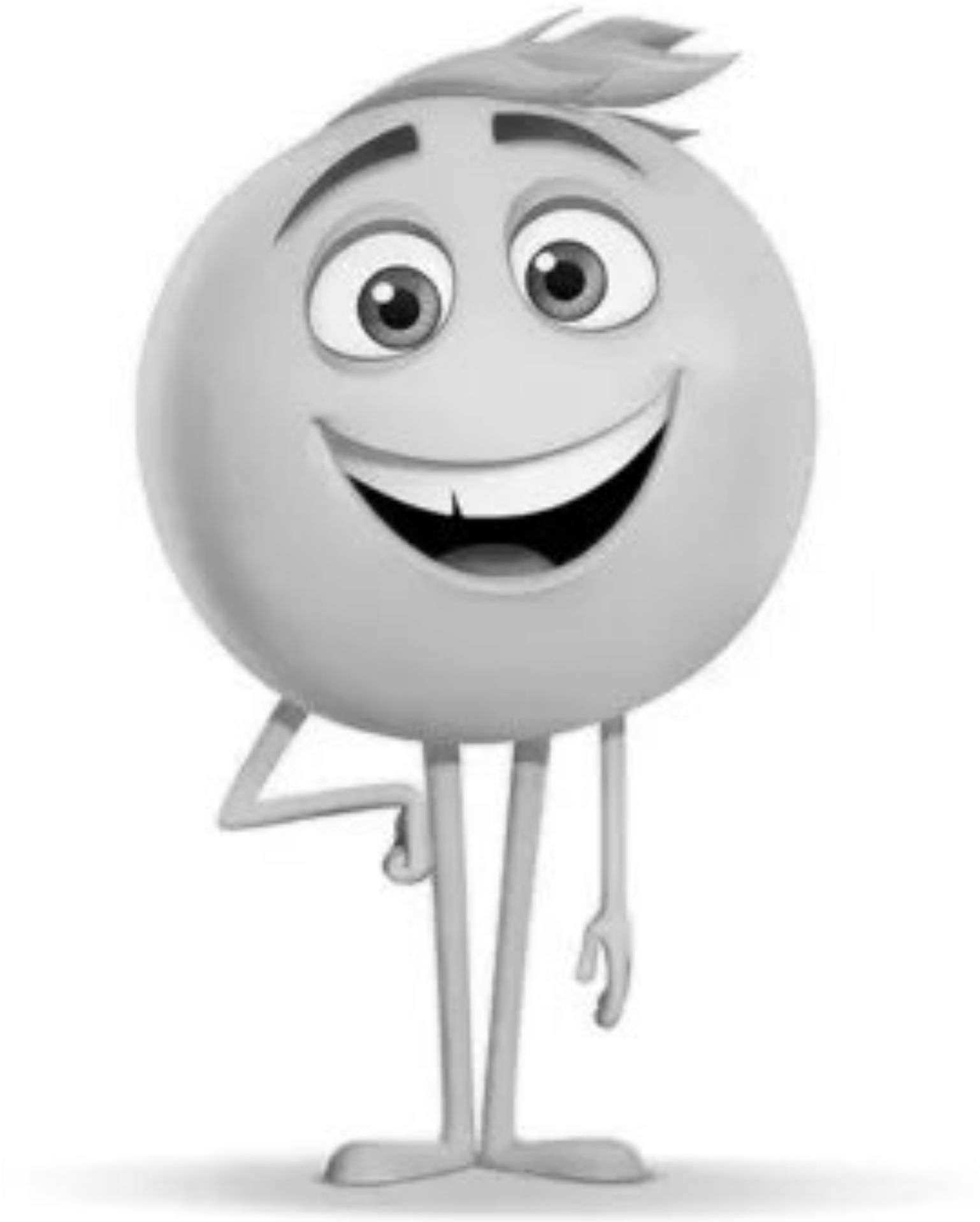
- * sme: identify a (niche) business model based on your OSS project
- * think big: investors and adequate sales and management teams



OSS LIFE - END OF THE DAY - SOMETIMES



OSS LIFE - END OF THE DAY





THANK YOU!

Daniel-Constantin Mierla

Co-Founder Kamailio Project

@miconda

asipto.com



Hoping For Another Berlin Edition In 2025

www.kamailioworld.com