PRETTY EASY PRIVACY

www: pep.foundation
mail: sva@pep.foundation
IRC: #PrettyEasyPrivacy on Freenode
twitter: @pepFoundation @sva
hashtags: #PrettyEasyPrivacy #PrivacyByDefault

p≡p

Privacy by Default.

Problem

Problem:

Online communication is visible like a postcard &
this world has mass surveillance

p≡p

Privacy by Default.

Problem & Solution

Problem:

Online communication is visible like a postcard &
this world has mass surveillance

Solution:

Mass Encryption == Privacy by Default.
Mass Anonymization == Privacy by Design.

p≡p
Privacy by Default.

Real-Time
General
Internet
Ops & Mgmt
Routing
Security
Transport
IRTF

New work
Chartering groups
BOFs

Other groups
Concluded groups
Non-WG lists

Documents
Search
Draft submission
Sign in to track docs

RFC streams
IAB
IRTF
ISE

Meetings
Agenda
Materials
Floor Plan
Past proceedings
Upcoming
Past
Request a session
Session requests

Other
IPR disclosures

Document	**Type**	Active Internet-Draft (individual)
	Last updated	2017-06-28
	Stream	(None)
	Intended RFC status	(None)
	Formats	plain text xml pdf html bibtex
Stream	**Stream state**	(No stream defined)
	Consensus Boilerplate	Unknown
	RFC Editor Note	(None)
IESG	**IESG state**	I-D Exists
	Telechat date	
	Responsible AD	(None)
	Send notices to	(None)

Email authors | IPR | ←References | →Referenced by | !Nits | Search lists

```
Network Working Group                                            V. Birk
Internet-Draft                                                 H. Marques
Intended status: Standards Track                                 Shelburn
Expires: December 31, 2017                                   pEp Foundation
                                                                 S. Koechli
                                                               pEp Security
                                                              June 29, 2017

             pretty Easy privacy (pEp): Privacy by Default
                          draft-birk-pep-00

Abstract

   Building on already available security formats and message transports
   (like PGP/MIME for email), pretty Easy privacy (pEp) describes
   protocols to automatize operations (key management, key discovery,
   private key handling including peer-to-peer synchronization of
   private keys and other user data across devices) that have been seen
   to be barriers to deployment of end-to-end secure interpersonal
   messaging.  pEp also introduces "Trustwords" (instead of
```

pretty Easy privacy (pEp): Privacy by Default
draft-birk-pep-00

Abstract

Building on already available security formats and message transports (like PGP/MIME for email), pretty Easy privacy (pEp) describes protocols to automatize operations (key management, key discovery, private key handling including peer-to-peer synchronization of private keys and other user data across devices) that have been seen to be barriers to deployment of end-to-end secure interpersonal messaging. pEp also introduces "Trustwords" (instead of fingerprints) to verify communication peers and proposes a trust rating system to denote secure types of communications and signal the privacy level available on a per-user and per-message level. In this document, the general design choices and principles of pEp are outlined.

https://datatracker.ietf.org/doc/draft-birk-pep/

p≡p
Privacy by Default.

What is p≡p?

...software for various platforms to easily use existing crypto tools (like GnuPG) ⇒ Pretty Easy

...designed to encrypt all digital written communication (with the starting point of email) ⇒ Privacy by Default.

...encrypts automatically with whatever (most privacy-enhancing) crypto standard available ⇒ Privacy by Default.

All end-user software must be hassle-free and zero-touch. ⇒ Pretty Easy

p≡p

Privacy by Default.

What is p≡p not?

...not yet-another-crypto-tool with closed user base.

...not a (centralized) platform provider.

...not implementing any own crypto.

...not replacing any existing crypto tool per se.

… not just an email encryption tool:
that's just the beginning \o/

p≡p

Privacy by Default.

Just the beginning...

We want to roll out **mass encryption**
to “optimize” the costs of mass surveillance!

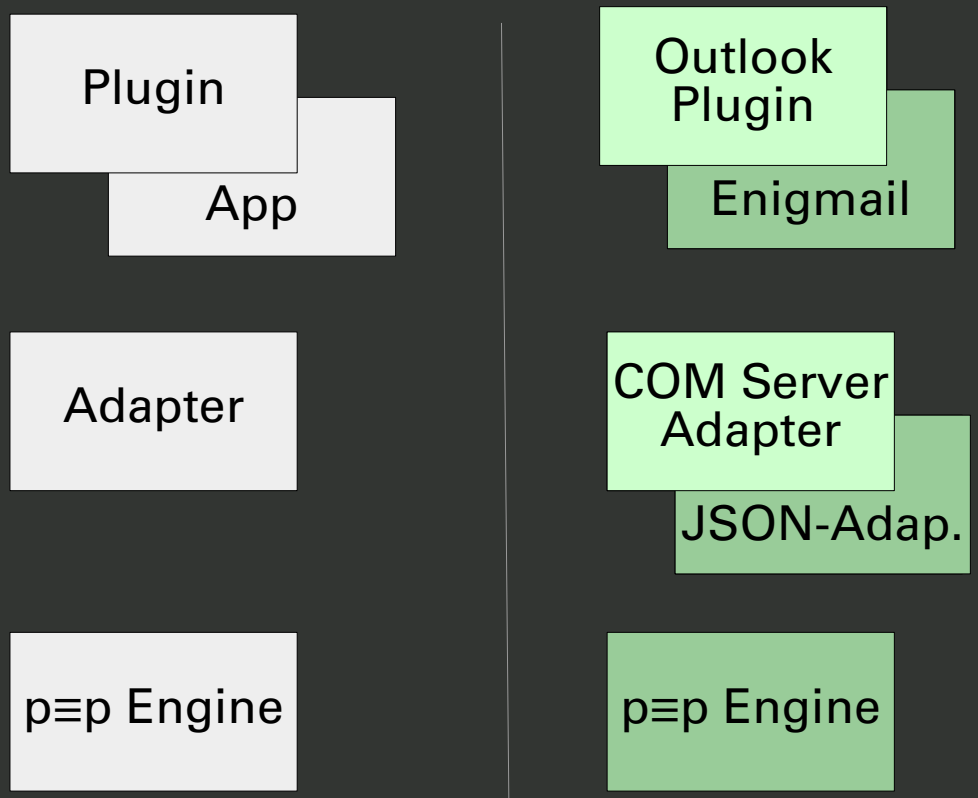
We want to make the use of crypto pretty easy:

The **developer plugs it** into apps.
The **user just uses it.**

By default.

p≡p
Privacy by Default.

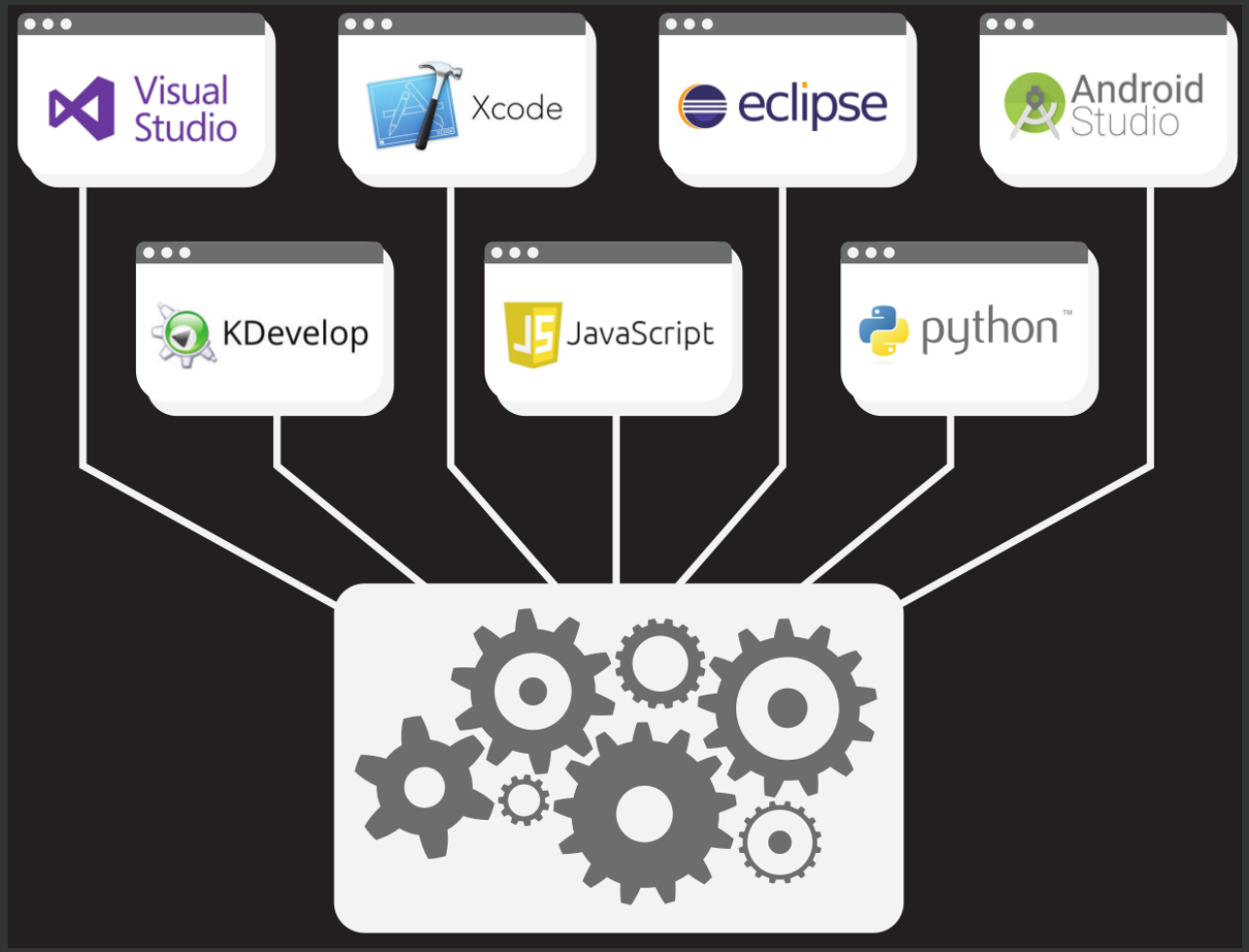
Architecture



p≡p

Privacy by Default.

Architecture



p≡p
Privacy by Default.

Adapter

… is a language/environment-specific interface between the engine API and an application development environment (like a programming language or IDE).

Basically adapters serve bindings.

Adapter	Example Languages
COM Server Adapter	C#, C++, VB.Net
JNI Adapter	Java (e.g. Android)
JSON Adapter	Javascript
ObjC Adapter	Swift (iOS, macOS)
Python Adapter	Python
C++/Qt Adapter	C++, Qt

p≡p
Privacy by Default.

And then?

Handles OpenPGP without hassle for the user:

Automatically encrypts
Encrypts the subject inline
Automatic key management
Import of existing keys
No keyserver or any other centralized infrastructure

Fingerprints ≡ Trustwords
Opt-in passphrase for keys
Disclaimer-Function
"Force-Protection"
"Passive-Mode"
Header encrypted & obfuscated
p≡pSync

p≡p
Privacy by Default.

Tryin’ to do everything right...

End-to-end encryption

Peer-to-peer transport

No centralized infrastructure
or closed services

Free Software with code audits

p≡p

Privacy by Default.

… and be compatible:

Multiple crypto technologies

Multiple message transports

Multiple platforms

Multiple languages

p≡p

Privacy by Default.

Repositories:

Android: https://**pep-security.lu/gitlab**/android/pep/

Outlook: https://**pep-security.lu**/dev/repos/pEp_for_Outlook/

iOS: https://**pep-security.ch**/dev/repos/pEp_for_iOS/

Enigmail: https://**sourceforge.net**/p/enigmail/source/ci/master/tree/

Engine & Adapter & MISC: https://pep.foundation/dev/

Everything: https://pep.foundation/pep-software

https://pep.software

p≡p

Privacy by Default.

Privacy by Default.

p≡p does what the user *would want to* do

Instead of writing how-to guides
we write user expectations
into software and protocols,

to automatize all steps a user would need to carry out.

⇒ Taking away "crypto needs" from users view (like *https)*

p≡p
Privacy by Default.

Conclusion

Users and Developers don't have to think
about the crypto anymore. They can just use it.

By default.

"It is this 'little hacker inside' that decides on the cryptography chosen to communicate with the message recipient."

https://pep.foundation – twitter@pepfoundation
e-mail: sva@pep.foundation – twitter@sva

p≡p
Privacy by Default.