

Concierge *de Code*

Le rêve de personne, la tâche de chacun.
(et comment Erlang peut aider)



Qui?

- ~ Learn You Some Erlang
- ~ Erlang Solutions Ltd.
- ~ BLOOM Digital Platforms
- ~ Erlang User of The Year 2012



Fred Hébert

Twitter: [@MononcQc](https://twitter.com/MononcQc)
Blog: <http://ferd.ca>



*La maintenance est le **prix à payer** pour pouvoir écrire du nouveau code.*

Temps dépensé

30%
dev

70%
maintenance

Temps dépensé



60% à 80% des coûts



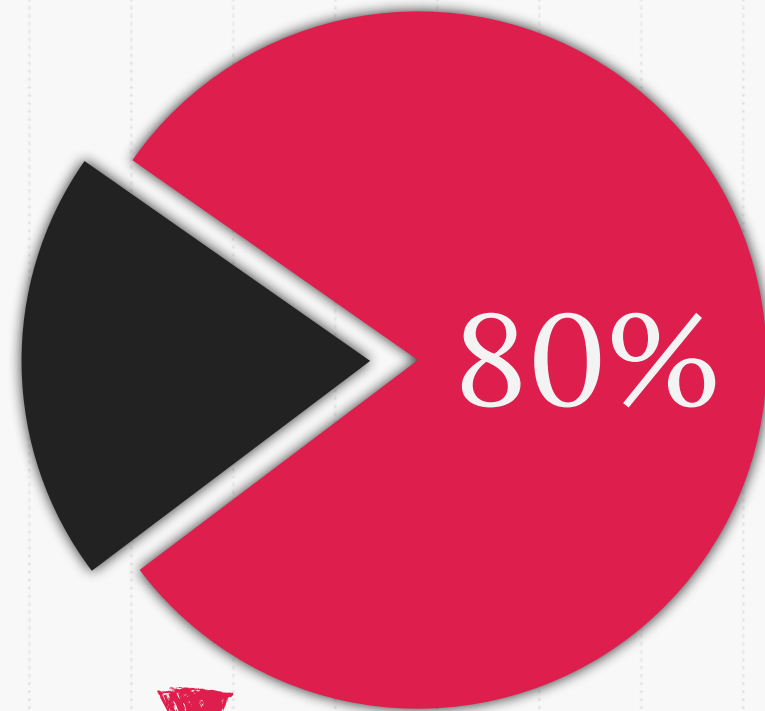
Types de maintenance

~ Corrective

~ Adaptative

~ Perfective

~ Urgences

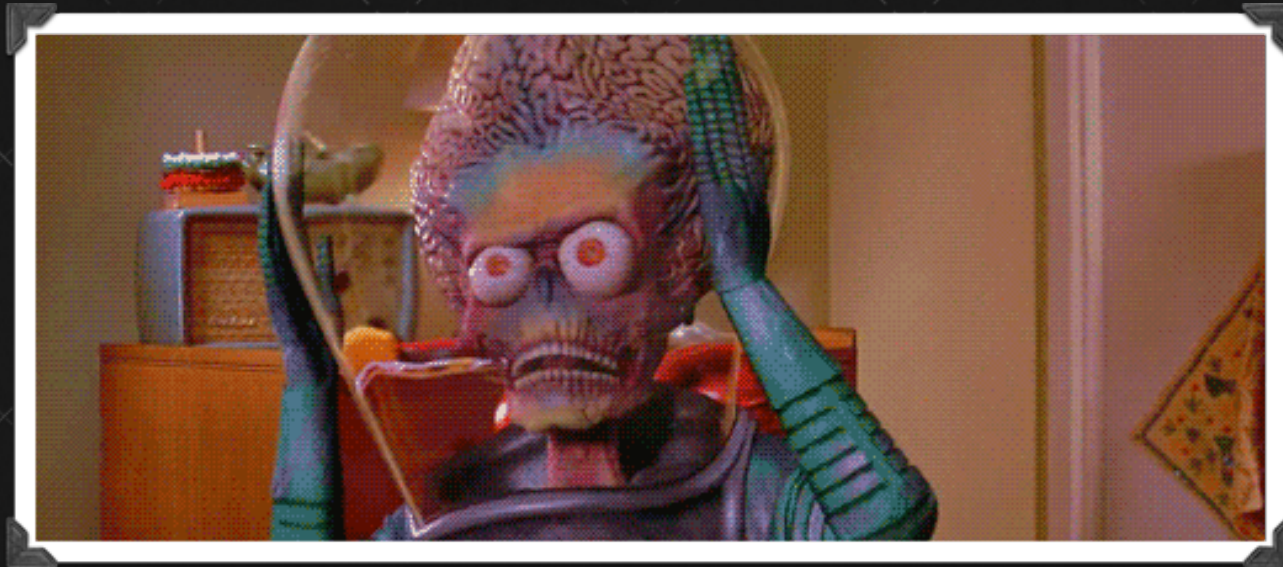


Lois de *Lehman*

- ~ Un programme qui est utilisé et dont l'implémentation représente une réalité donnée, subit un changement continu ou devient de moins en moins utile. Le processus de changement ou de perte de qualité se continue jusqu'à ce qu'il soit jugé plus pratique de remplacer le système par un nouveau.

Lois de *Lehman*

- ~ Alors qu'un programme est continuellement modifié, sa complexité, tout comme la détérioration de sa structure, augmente à moins que du travail soit fait pour la maintenir ou la réduire.



Le problème de complexité est
humain.

Conséquences?



22.7% à 57.6% de tout le temps d'un projet est utilisé pour tenter de comprendre le système.

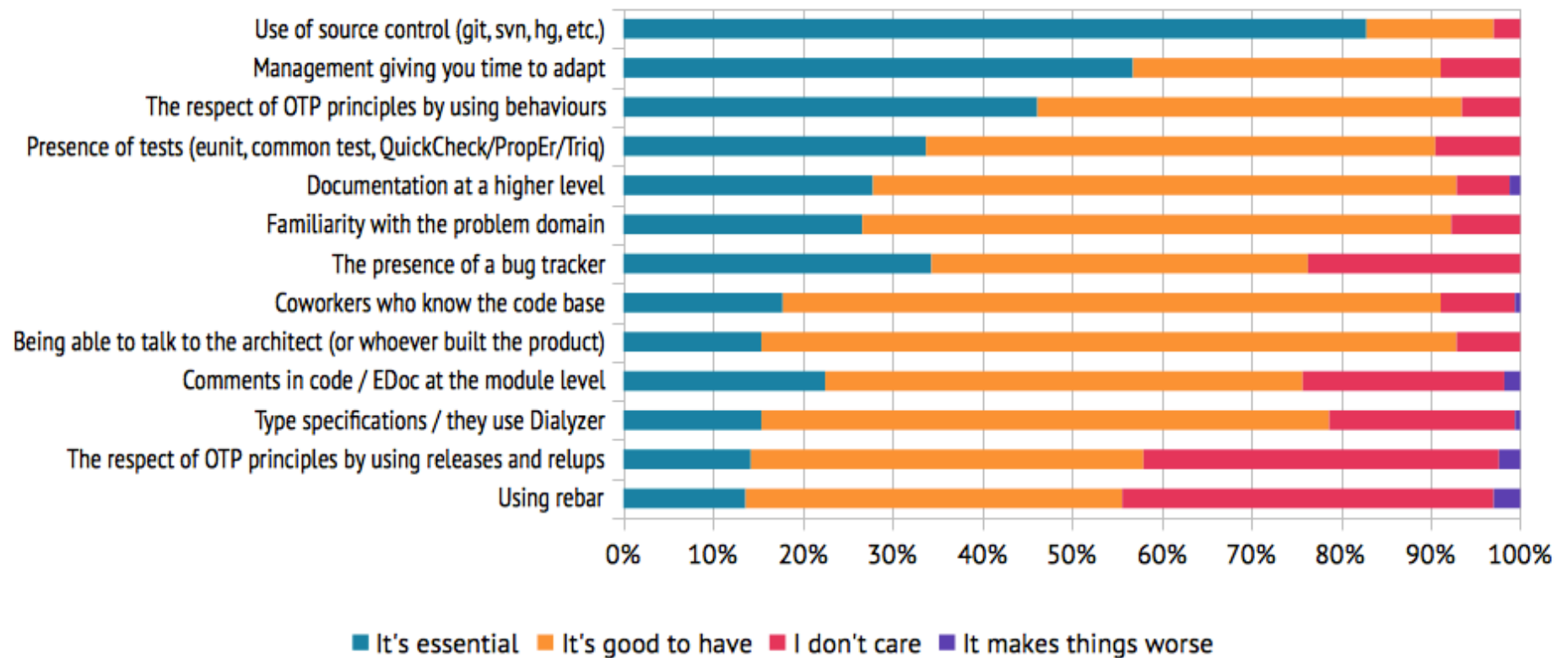
Sources d'information

- ~ Code
- ~ Collègues
- ~ Outils de données (bug trackers, etc.)
- ~ Documentation



***Erlang** peut aider*

*You are hired to take over an existing Erlang code base.
Based on your experience, what do you think is important
for you to feel comfortable and take 'ownership' of the code?*



OTP

L'importance des *behaviours*



Composantes

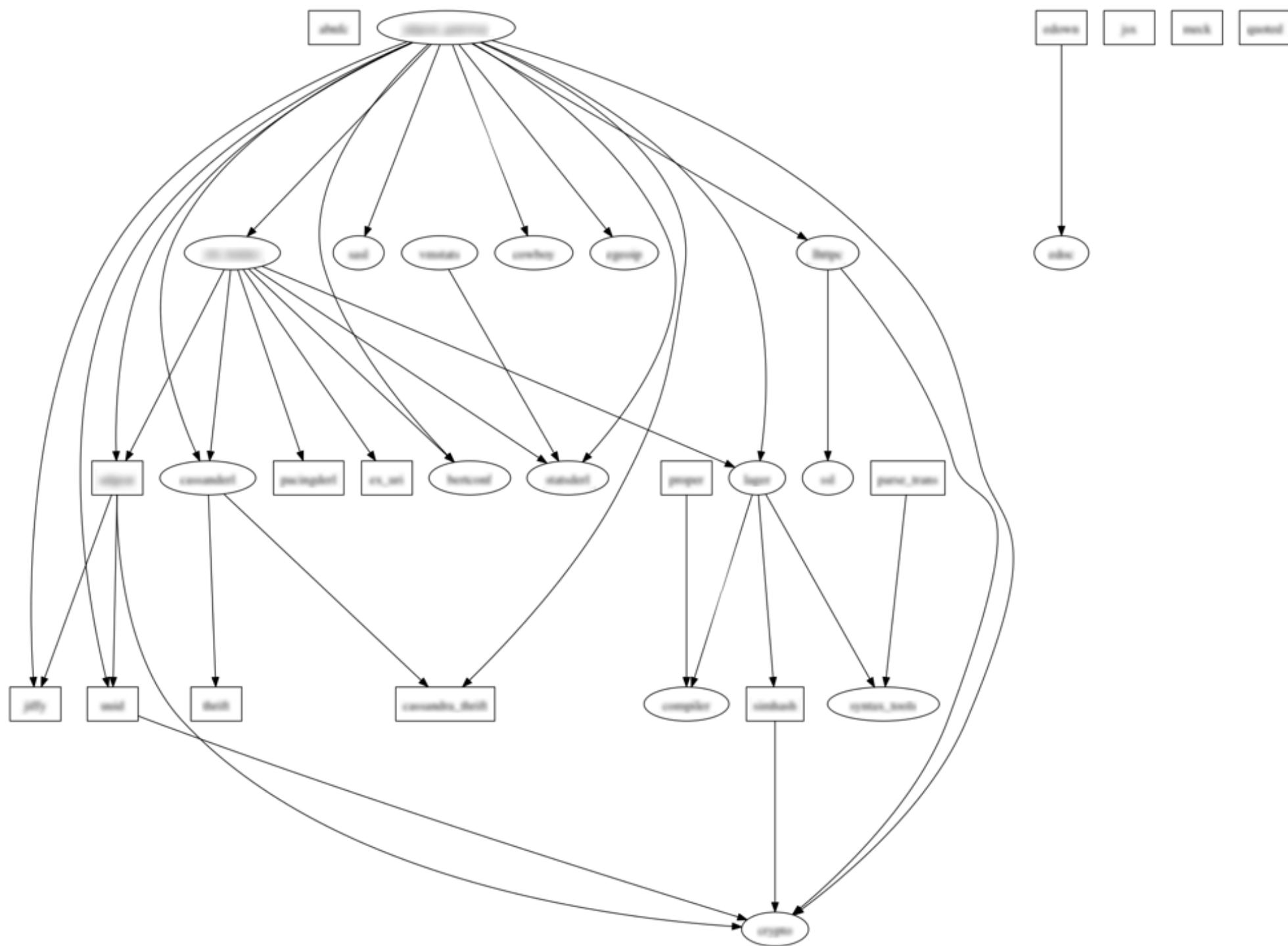
- ~ Workers
- ~ Superviseurs
- ~ Applications



De *Jenga* à Lego

Protocoles

- ~ Isolation
- ~ Communications par messages
- ~ Définir des standards pour structurer l'abstrait (HTTP, TCP/IP, BitTorrent Wire)
- ~ Applications communiquent via protocoles internes et OTP

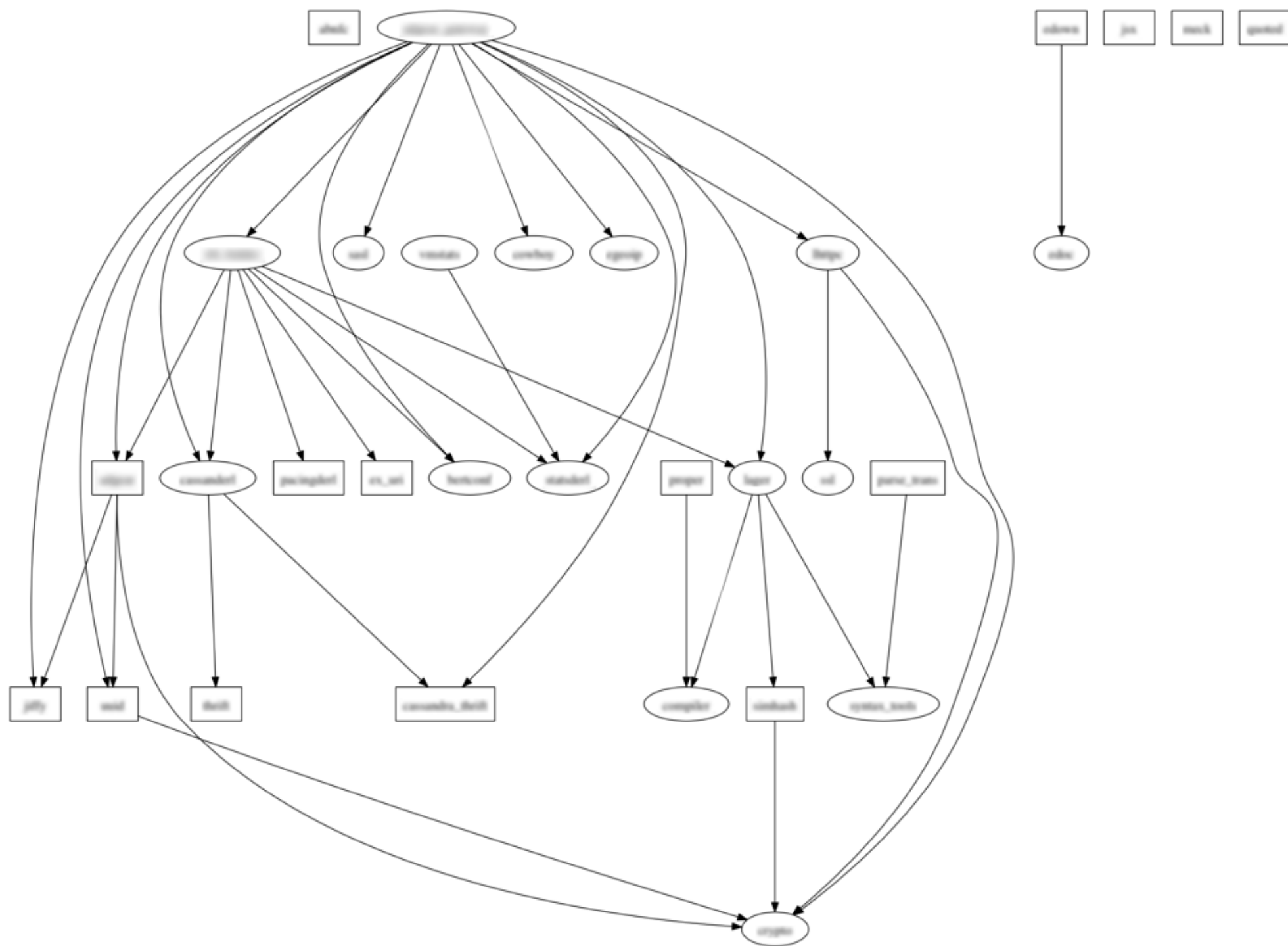




*Modifications **sans** Erlang/OTP*

*Comprendre, de **loin**.*

- ~ Utilisé dans tous les domaines
- ~ Compréhension de structure rapide
- ~ Metadata d'applications



Patterns standards

- ~ serveurs
- ~ automates finis (FSMs)
- ~ event handlers
- ~ superviseurs



*Patterns dans **d'autres** langages*



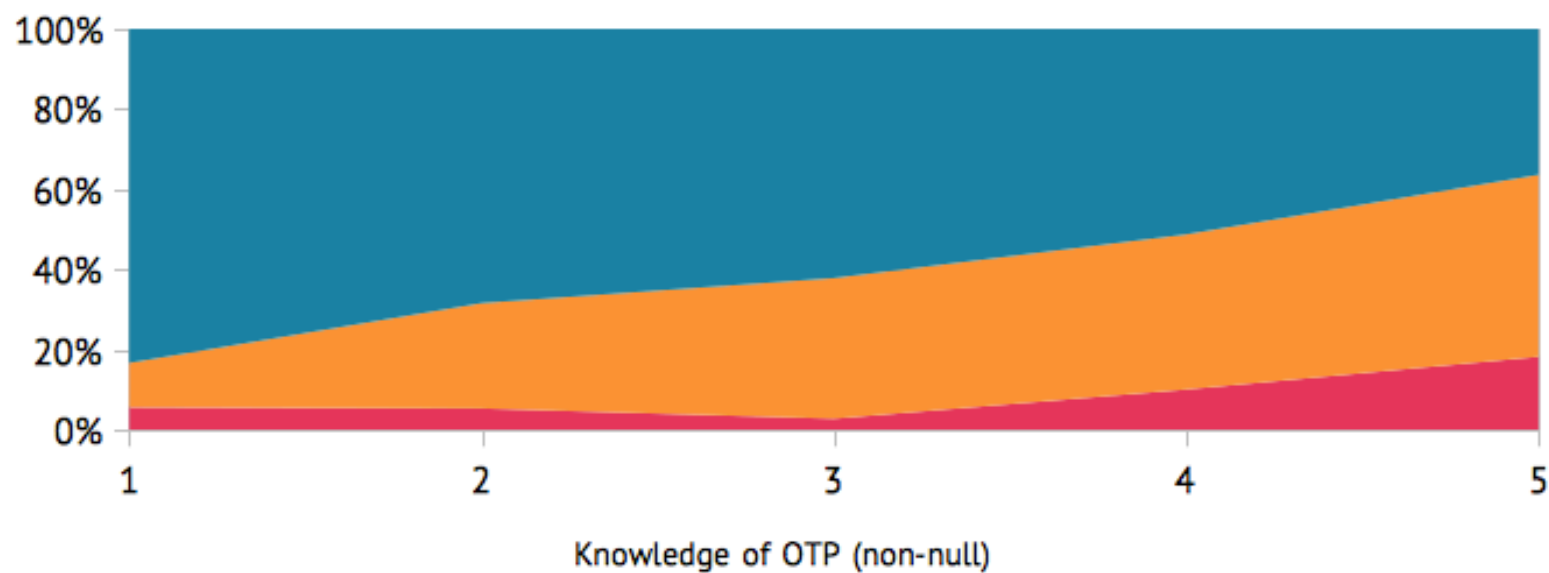
Tout le monde s'en sert!

Reverse *Engineering* ++

- ~ Tracing!
- ~ `sys:get_status(Worker)`
- ~ `sys:trace(Worker, true)`
- ~ DBG

Importance of time given by management to adapt, over knowledge of OTP

■ It makes things worse ■ I don't care ■ It's good to have ■ It's essential



Problèmes

- ~ Conflits de versions de librairies
- ~ Pas de *namespaces*
- ~ Package management
- ~ i18n, l10n



Les Systèmes **OTP** sont **Solides**

Learn You Some **Erlang** for Great Good!

A Beginner's Guide



Fred Hébert



coupon **PREORDER** = -30%
nostarch.com/erlang