Privacy Engineering and the Agile Turn

Seda Gürses
seda@esat.kuleuven.be
COSIC, University of Leuven
CITP, Princeton University

28. Februar 2017
SecAppDev

getting privacy engineering right?

getting privacy engineering right?

privacy research

software engineering practice

privacy research

software engineering practice

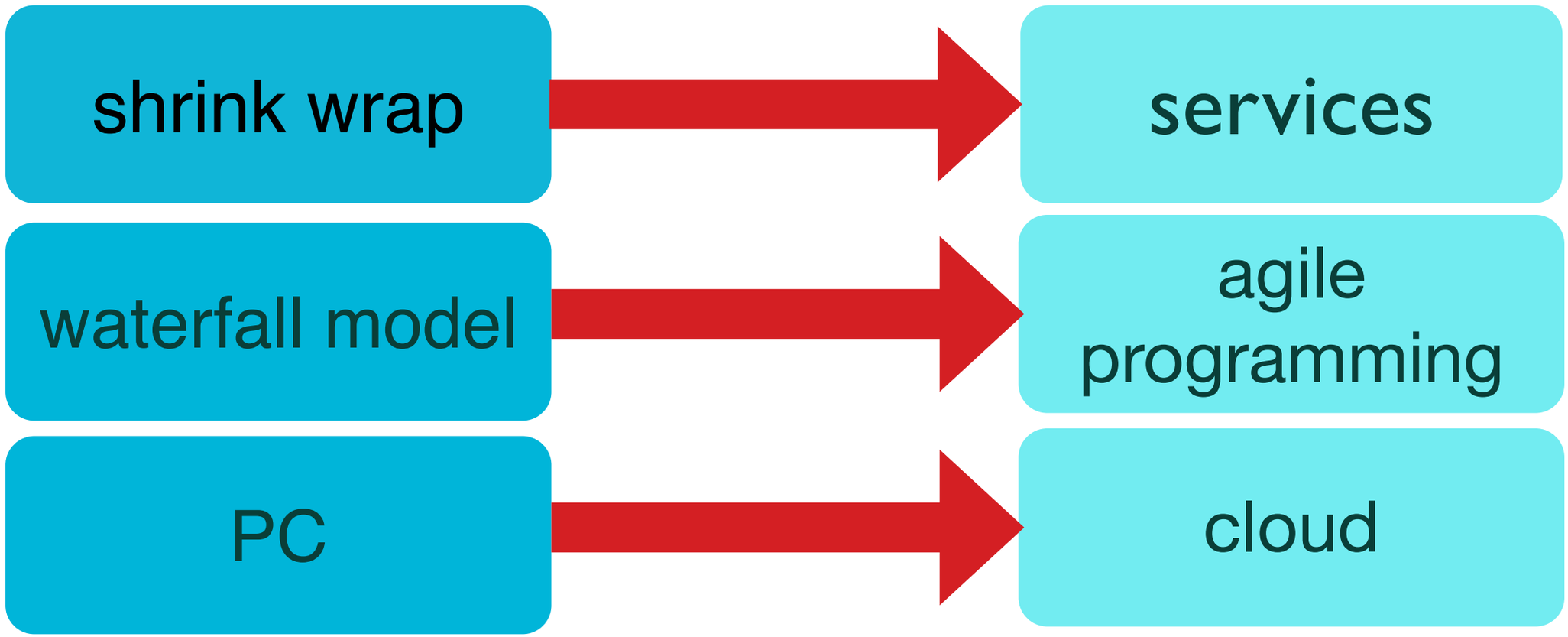
can it be that the practices around the production of software are an important element of privacy research?

privacy research

software engineering practice

matters?

the turn to agile



what is the impact of

the turn to agile in software engineering practice

on computer science research in privacy?

PRIVACY RESEARCH PARADIGMS

privacy as confidentiality

privacy as control

privacy as practice

privacy theory

methods

techniques

tools

what is the impact of

the turn to agile in software engineering practice

on computer science research in privacy?

methodology

- exploratory study (work in progress)
  - develop and shape an agenda for further study
- interviews and chats
  - devs, devops, product managers, a/b testers, AI/data product developers, data engineers, privacy officers
- industry white papers
- legal and policy literature

shrink wrap software

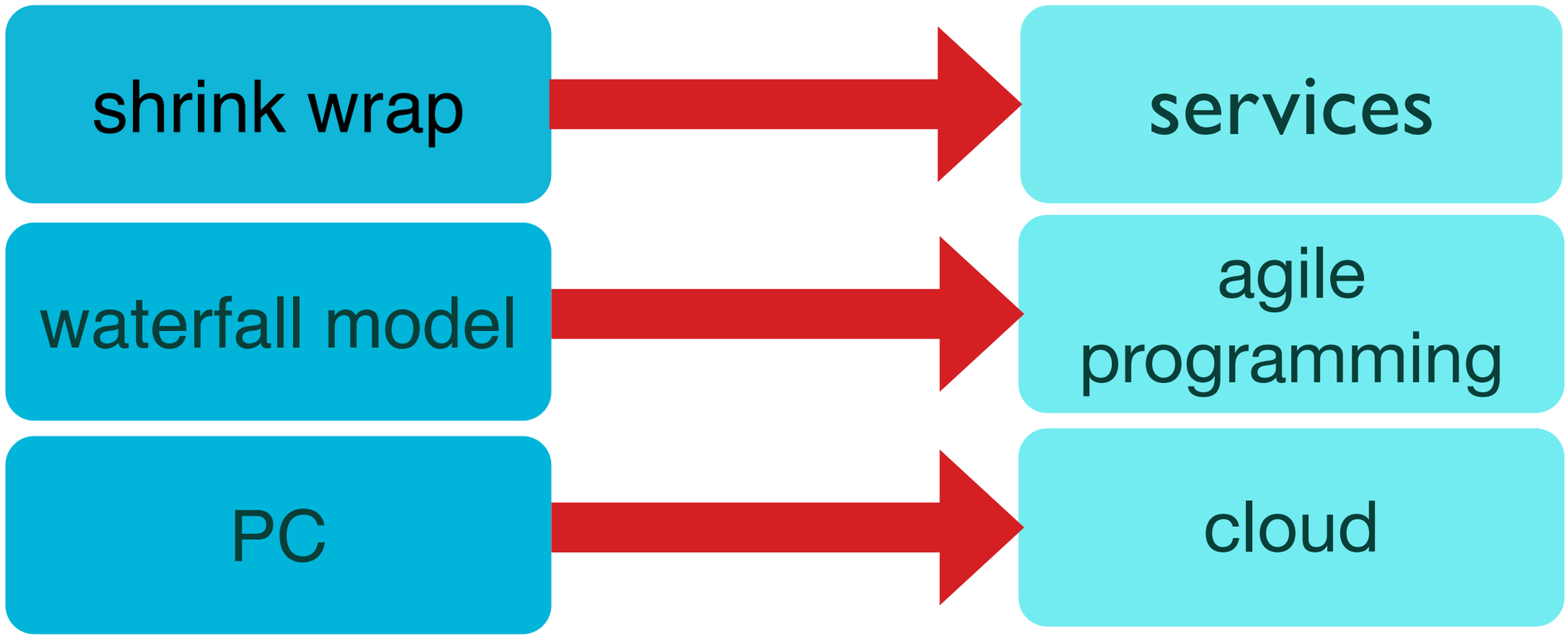
cloud
IaaS/PaaS

SOA

SaaS

agile methods

the turn to agile




shrink wrap
services

1) All teams will henceforth expose their data and functionality through service interfaces.

2) Teams must communicate with each other through these interfaces.

3) There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.

4) It doesn't matter what technology they use. HTTP, Corba, Pubsub, custom protocols – doesn't matter. Bezos doesn't care.

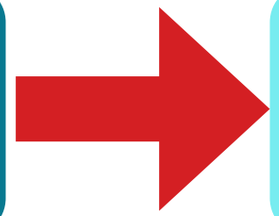
5) All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.

6) Anyone who doesn't do this will be fired.
~2001/2002

shrink wrap	enterprise	apps	services
binary runs solely on client side			server (thin) client model
requires matching soft & hardware			data “secured” by service
updates & maintenance cumbersome			updates and maintenance server side
user has control (oh no!)			collaborative
pay in advance			pay as you use/trial
Microsoft Word			office 365

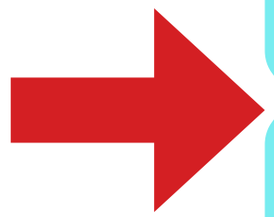
implications of the shift to services

server - thin client model



transaction throughout use

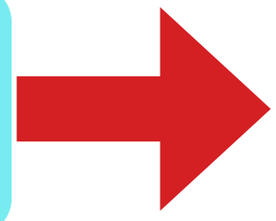
bundled services



agile service integration

pooling of data

licensing and pricing models



intensified tracking

shrink wrap
software production

version
+
purchase

use

time

service bundle

pay per use

use

team integration | SDK/PaaS | cybersecurity | performance

CRM | data brokers | analytics | AB Testing | UX capture

production tools

advertisement

authentication | payment | maps | social | embedded media

picture album creation service

fullstory

FEATURES PRICING CUSTOMERS COMPANY BLOG SUPPORT

FREE 14-DAY TRIAL LOG IN

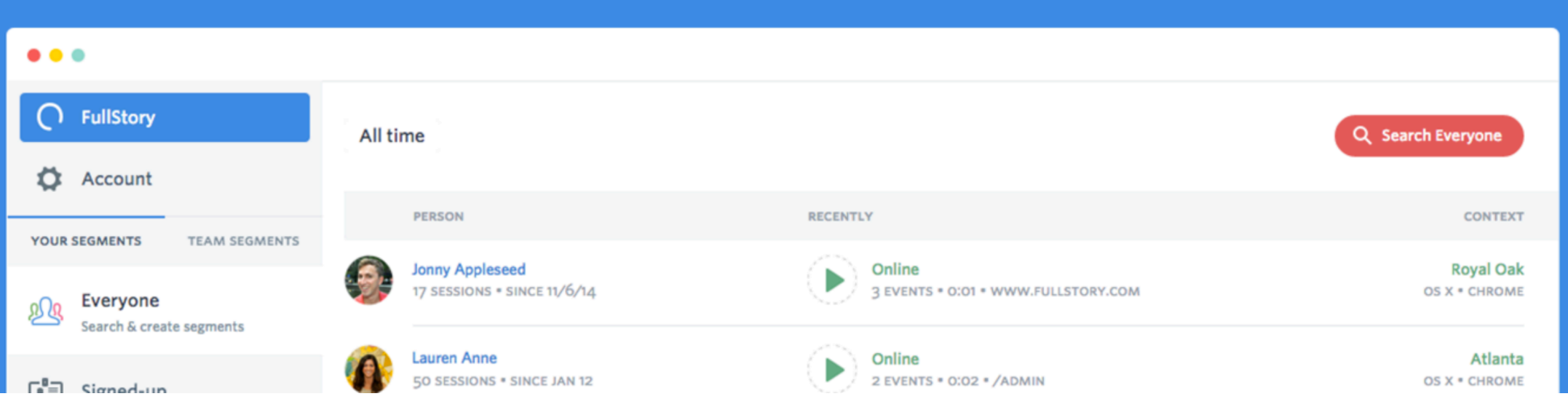
See what your users see.

FullStory lets your company easily record, replay, search, and analyze each user's actual experience with your website. Think of it as your team's super-searchable DVR for all customer interactions.

Start your free 14-day trial today!

you@widgetco.com LET'S GO!

Watch the video (1:13)



fullstory in top 1 million sites

http://uservoice.com

http://sproutvideo.com

http://startapp.com

http://fitocracy.com

http://remitly.com

http://wahoofitness.com

http://wayup.com

http://tieks.com

http://moosejaw.com

http://clickminded.com

http://keen.io

http://samcart.com

http://meuspedidos.com.br

http://oyorooms.com

http://urbanclap.com

http://himalayastore.com

http://travelport.com

http://credomobile.com

http://deputy.com

http://referralcandy.com

http://codeschool.com

http://owler.com

http://surfdome.com

http://autopilothq.com

http://conte.it

http://autoeurope.com

http://thebouqs.com

http://mymove.com

http://scripted.com

http://namely.com

http://shethinx.com

http://castorama.pl

http://nexojornal.com.br

Thanks to Dillon Reisman from Princeton U. for the web crawl


waterfall model
agile
programming

waterfall model

spiral model

agile programming

Xtreme programming

waterfall model

requirements analysis and specification

architectural design

implementation and integration

verification

operation and maintenance

agile manifesto

individuals and interactions	process and tools
working software	comprehensive documentation
customer collaboration	contract negotiation
responding to change	following a plan

eXtreme Programming

if short iterations are good, make them as short as possible

if simplicity is good, do the simplest thing that can work

if testing is good, test all the time

if code reviews are good, review code continuously

implications of the shift to agile dev

testing testing testing → user centric development

data centric development

short iterations → rapid feature development

simplicity → reuse and modularity

server - thin client model

feature inflation

customer

product manager

behavioral analytics

rapid feature development

where do features come from?

where do features go?

boss/VC said so

designers said so

competitor did it

data centric development

anecdotes

data products

metrics

user/behavioral analytics

data centric development

predictive modeling 4 pricing

user churn

new information panel

website

perspective 3: behavior and data centricity

- recursively keeping track:
  - capturing behavior of users
  - capturing behavior of service components
  - capturing behavior of your capture models
    - QA and continuous monitoring become one thing

time

feature space

service bundle

pay per use

consent

use

how are shifts in software engineering and the ecosystem relevant to privacy research and practice?

Philip Agre: Two models of privacy

These systems capture knowledge of people's behavior, and they reconfigure them through rapid development of features that are able to identify, sequence, reorder and transform human activities.

This also means that they open these human activities to evaluation in terms of economic efficiency.
Philip Agre.

Moving Targets: Security and Rapid-Release in Firefox

Sandy Clark
saender@cis.upenn.edu
University of Pennsylvania

Michael Collis
mcollis@cis.upenn.edu
University of Pennsylvania *

Matt Blaze
mab@crypto.com
University of Pennsylvania

Jonathan M. Smith
jms@cis.upenn.edu
University of Pennsylvania

can't apply security frameworks

no threat modeling

no risk assessment

code maturity? lol

rapid feature development

++ vulnerability density

++ immature code

honeymoon

defies attackers learning curve

how are shifts in software engineering and the ecosystem relevant to privacy research?

impact of the agile turn?

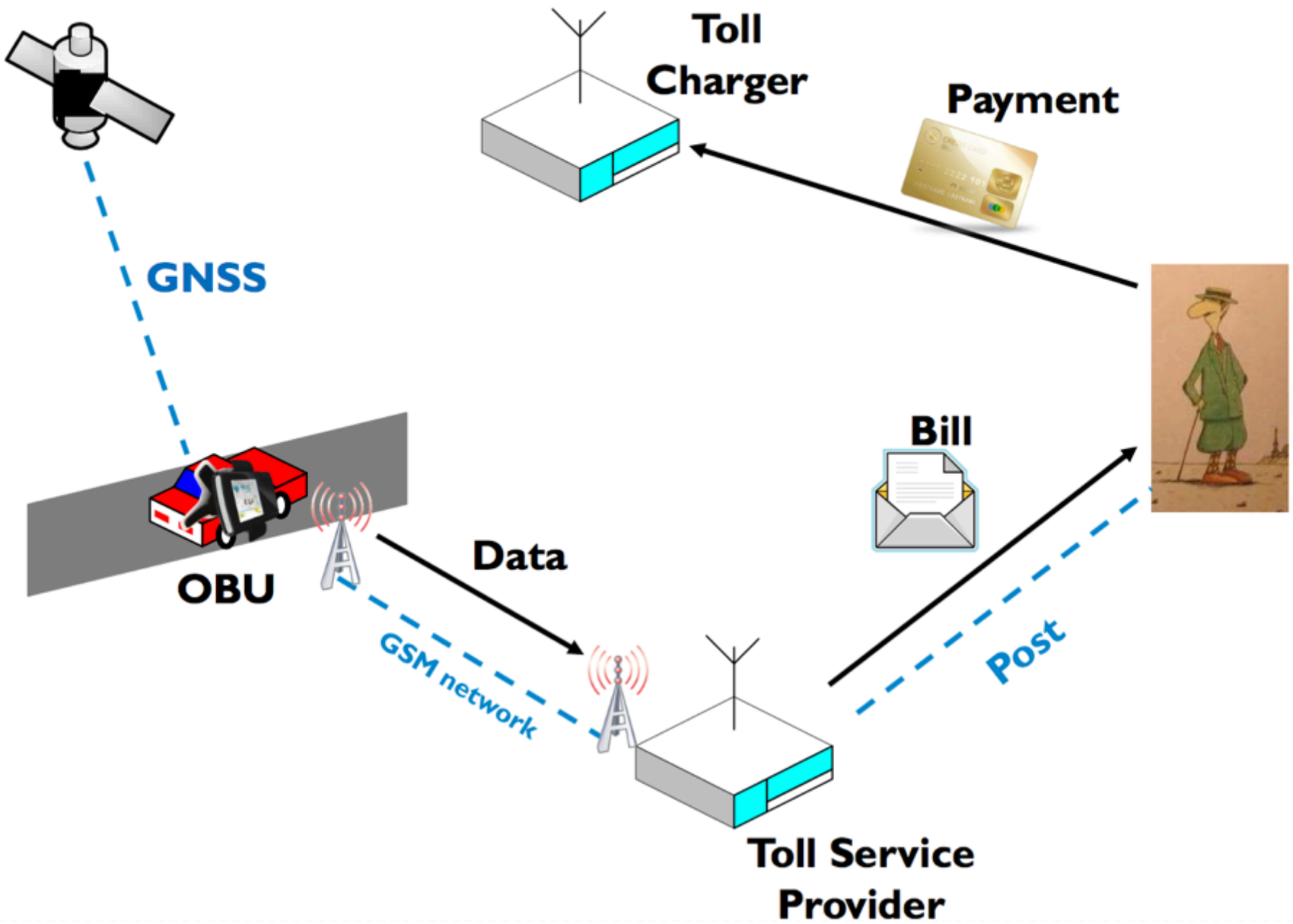
privacy as confidentiality

data minimization

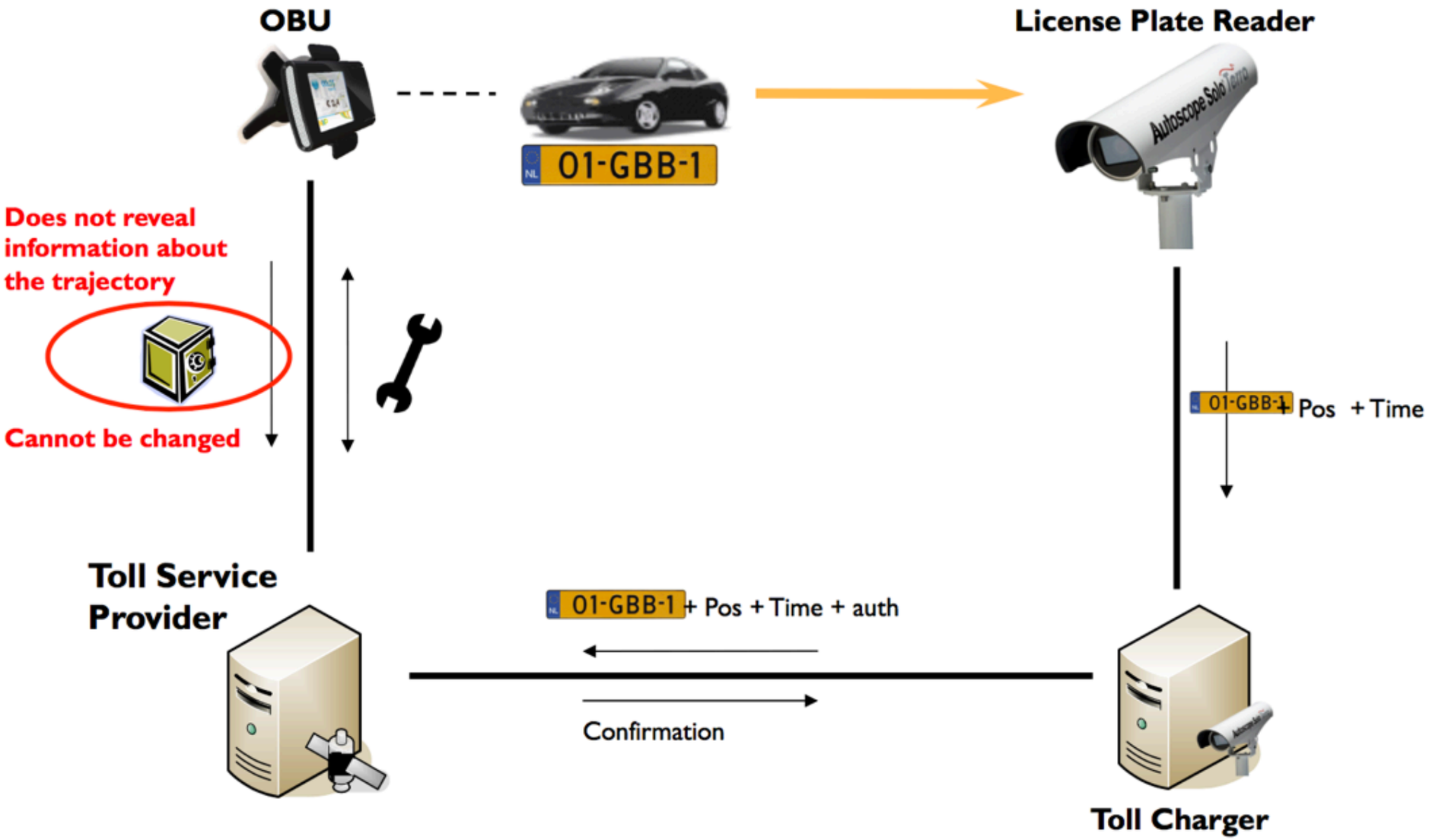
properties with mathematical guarantees

avoid single point of failure

EETS straightforward implementation



How does it work?



Data Minimization strategies

well-defined goal

reference system

privacy requirements

- minimize collection
- minimize disclosure
- minimize replication
- minimize centralization
- minimize linkability
- minimize retention

SOK: Secure Messaging (Unger et al.)

well-defined goal	(interoperable/federated) secure messaging
	trust establishment
	conversation security
	transport privacy
privacy requirements	confidentiality + perfect forward/backward secrecy
	message/participation deniability
	anonymity ...
threat model (adversary)	local/global/ISP...
other quality requirements	usability and adoption

CONVERSATION SECURITY PROTOCOLS AND THEIR USABILITY AND ADOPTION IMPLICATIONS. NO APPROACH REQUIRES ADDITIONAL USER EFFORT.

Scheme	Example	Security and Privacy													Adoption					Group Chat				
		Confidentiality	Integrity	Authentication	Participant Consistency	Destination Validation	Forward Secrecy	Backward Secrecy	Anonymity Preserving	Speaker Consistency	Causality Preserving	Global Transcript	Message Unlinkability	Message Repudiation	Particip. Repudiation	Out-of-Order Resilient	Dropped Message Resilient	Asynchronicity	Multi-Device Support	No Additional Service	Computational Equality	Trust Equality	Subgroup Messaging	Contractable	Expandable
TLS+Trusted Server†*	Skype	-	-	-	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	-	●	●	●	◐	◐
Static Asymmetric Crypto†*	OpenPGP, S/MIME	●	●	●	-	-	-	-	●	-	-	-	-	-	-	●	●	●	●	●					
+IBE†	Wang et al.	-	●	●	-	-	-	-	●	-	-	-	-	-	-	●	●	●	●	-					
+Short Lifetime Keys	OpenPGP Draft	●	●	●	-	-	◐	◐	●	-	-	-	-	-	-	●	●	●	●	-					
+Non-Interactive IBE†	Canetti et al.	●	●	●	-	-	●	-	●	-	-	-	-	-	-	◐	●	●	●	●					
+Puncturable Encryption†	Green and Miers	●	●	●	-	-	●	-	●	-	-	-	-	-	-	●	●	●	●	●					
Key Directory+Short Lifetime Keys†	IMKE	●	●	◐	-	●	◐	◐	-	-	-	-	●	●	●	●	●	-	-	-					
+Long-Term Keys†	SIMPP	●	●	◐	-	●	◐	◐	-	-	-	-	●	●	-	●	●	-	-	-					
Authenticated DH†*	TLS-EDH-MA	●	●	●	●	●	◐	◐	●	-	-	-	●	●	◐	●	●	-	-	●					
+Naïve KDF Ratchet*	SCIMP	●	●	●	●	●	●	◐	●	◐	-	-	●	●	◐	◐	◐	-	-	●					
+DH Ratchet†*	OTR	●	●	●	●	●	◐	●	●	◐	◐	-	●	●	◐	◐	◐	-	-	●					
+Double Ratchet†*	Axolotl	●	●	●	●	●	●	●	●	◐	◐	-	●	●	◐	●	◐	-	-	●					
+Double Ratchet+3DH AKE†*	-	●	●	●	●	●	●	●	◐	◐	◐	-	●	●	●	●	◐	-	-	●					
+Double Ratchet+3DH AKE+Prekeys†*	TextSecure	●	●	●	●	●	●	●	-	◐	●	-	●	●	●	◐	◐	●	-	-					
Key Directory+Static DH+Key Transport†	Kikuchi et al.	●	●	-	-	●	◐	◐	-	-	-	-	●	●	-	●	●	●	-	-	-	-	-	●	●
+Authenticated EDH+Group MAC†	GROK	●	●	◐	-	●	◐	◐	●	-	-	-	●	●	-	●	●	●	-	-	-	-	-	●	●
GKA+Signed Messages+Parent IDs†	OldBlue	●	●	●	●	●	◐	◐	●	●	-	-	-	-	-	●	●	◐	-	●	●	●	-	-	-
Authenticated MP DH+Causal Blocks†*	KleeQ	●	●	◐	◐	◐	●	●	●	◐	●	-	●	●		●	●	◐	-	●	●	●	-	-	●
OTR Network+Star Topology†	GOTR (2007)	●	●	-	-	-	◐	●	-	-	-	-	●	●		◐	●	◐	-	●	-	-	-	●	●
+Pairwise Topology†		●	●	●	●	●	◐	●	●	-	-	-	●	●		◐	●	◐	-	●	●	●	●	●	●
+Pairwise Axolotl+Multicast Encryption*	TextSecure	●	●	●	-	●	●	●	-	●	●	-	●	●		●	●	●	-	-	●	●	●	●	●
DGKE+Shutdown Consistency Check†	mpOTR	●	●	●	●	●	◐	◐	●	◐	-	-	-	●		●	●	-	-	●	●	●	-	-	-
Circle Keys+Message Consistency Check†	GOTR (2013)	●	●	●	●	●	◐	◐	●	●	●	●	●	●		●	-	-	-	◐	●	●	-	●	●

SOK: Secure Messaging (Unger et al.)

legacy software not made with E2E security in mind

unsolved problems: e.g., group chat

solved problems: not applied

current implementations: proprietary/no specification

Reflections: The ecosystem is moving

moxie0 on 10 May 2016

Software exists as part of an ecosystem, and **the ecosystem is moving**. The platform changes out from under it, the networks evolve, security threats and countermeasures are in constant shift, and the collective UX language rarely sits still. As more money, time, and focus has gone into the ecosystem, the faster the whole thing has begun to travel.

One of the controversial things we did with Signal early on was to build it as an unfederated service. Nothing about any of the protocols we've developed requires centralization; it's entirely possible to build a federated Signal Protocol based messenger, but I no longer believe that it is possible to build a *competitive* federated messenger at all.

impact of the agile turn?

privacy as control

data protection/FIPPS compliance

transparency and accountability

Bell Group

information we collect	ways we use your information				information sharing	
	to provide service and maintain site	marketing	telemarketing	profiling	other companies	public forums
contact information	■	opt in		■	opt out	
cookies	■		■	■		
demographic information	■	opt in		■	opt out	
financial information	■					
health information					■	
preferences	■		■		■	
purchasing information	■	opt in		■	opt out	
social security number & gov't ID						
your activity on this site	■	opt in	■	■	opt out	
your location		■		■		

Access to your information
This site gives you access to your contact data and some of its other data identified with you

How to resolve privacy-related disputes with this site
Please email our customer service department

bell.com
5000 Forbes Avenue
Pittsburgh, PA 15213 United States
Phone: 800-555-5555
help@bell.com

impact of the agile turn?

privacy as practice

privacy integral to collective info practices

improve user agency in negotiating privacy

transparency of social impact

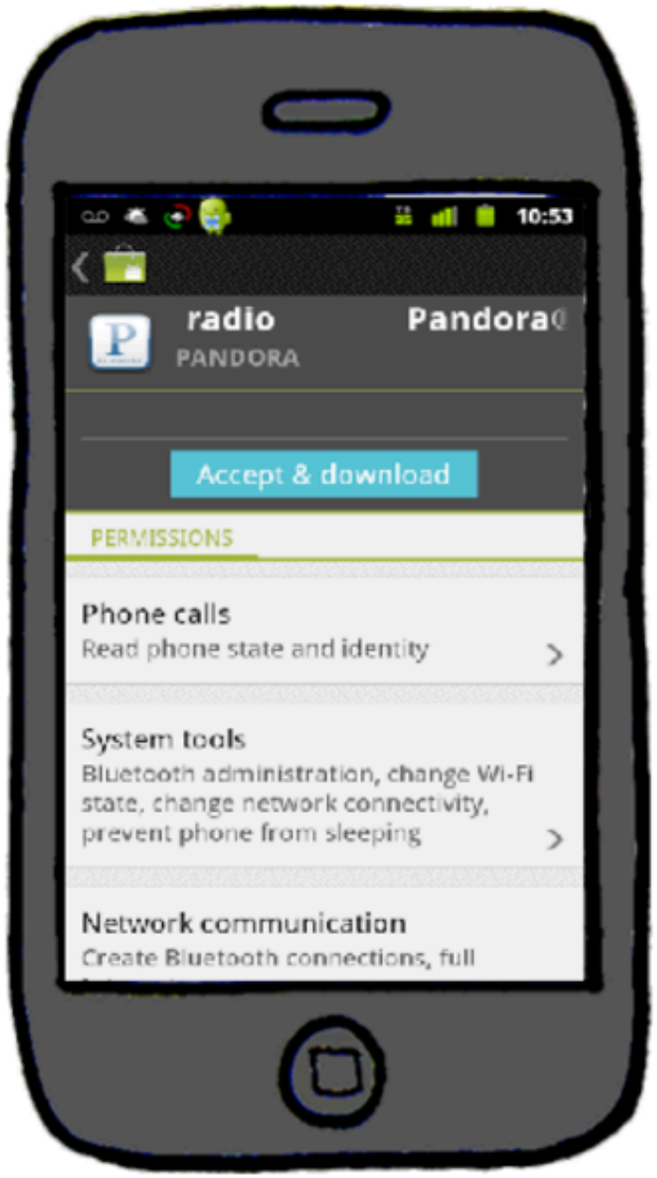
Android Permissions: User attention, comprehension, and Behavior (Felt et al., 2012)

Permission	*n*	Options	Responses	
`INTERNET` Category: Network communication Label: Full Internet access	109	Send information to the application's server	45	41.3%
		Load advertisements	30	27.5%
		None of these	16	14.7%
		Read your text messages	13	11.9%
		Read your list of phone contacts	11	10.1%
		I don't know	36	33.0%
`READ_PHONE_STATE` Category: Phone calls Label: Read phone state and identity	85	Read your phone number	41	47.7%
		See who you have called	37	43.0%
		Track you across applications	20	23.3%
		Load advertisements	11	12.8%
		None of these	10	11.6%
		I don't know	15	17.4%
`CALL_PHONE` Category: Services that cost you money Label: Directly call phone numbers	83	Place phone calls	30	35.3%
		Charge purchases to your credit card	27	31.8%
		None of these	16	18.8%
		See who you have made calls to	14	16.5%
		Send text messages	11	12.9%
		I don't know	16	18.8%
`WRITE_EXTERNAL_STORAGE` Category: Storage Label: Modify/delete SD card contents	92	Read other applications' files on the SD card	41	44.6%
		Change other applications' files on the SD card	39	42.4%
		None of these	16	17.4%
		See who you have made phone calls to	15	16.3%
		Send text messages	11	12.0%
		I don't know	15	16.3%
`WAKE_LOCK` Category: System tools Label: Prevent phone from sleeping	81	Keep your phone's screen on all the time	49	60.5%
		Drain your phone's battery	37	45.7%
		None of these	7	8.6%
		Send text messages	4	4.9%
		Delete your list of contacts	4	4.9%
		I don't know	13	16.0%
`CHANGE_NETWORK_STATE`		Turn your WiFi on or off	36	52.9%
		Send information to the application's server	13	19.1%

Android Permissions: User attention, comprehension, and Behavior (Felt et al., 2012)

Permission	*n*	Options	Responses	
`INTERNET` Category: Network communication Label: Full Internet access	109	✔ Send information to the application's server ✔ Load advertisements ✗ None of these ✗ Read your text messages ✗ Read your list of phone contacts *I don't know*	45 30 16 13 11 36	41.3% 27.5% 14.7% 11.9% 10.1% 33.0%
`READ_PHONE_STATE` Category: Phone calls Label: Read phone state and identity	85	✔ Read your phone number ✗ See who you have called ✔ Track you across applications ✗ Load advertisements ✗ None of these *I don't know*	41 37 20 11 10 15	47.7% 43.0% 23.3% 12.8% 11.6% 17.4%
`CALL_PHONE` Category: Services that cost you money Label: Directly call phone numbers	83	✔ Place phone calls ✗ Charge purchases to your credit card ✗ None of these ✗ See who you have made calls to ✗ Send text messages *I don't know*	30 27 16 14 11 16	35.3% 31.8% 18.8% 16.5% 12.9% 18.8%
`WRITE_EXTERNAL_STORAGE` Category: Storage Label: Modify/delete SD card contents	92	✔ Read other applications' files on the SD card ✔ Change other applications' files on the SD card ✗ None of these ✗ See who you have made phone calls to ✗ Send text messages *I don't know*	41 39 16 15 11 15	44.6% 42.4% 17.4% 16.3% 12.0% 16.3%
`WAKE_LOCK` Category: System tools Label: Prevent phone from sleeping	81	✔ Keep your phone's screen on all the time ✔ Drain your phone's battery ✗ None of these ✗ Send text messages ✗ Delete your list of contacts *I don't know*	49 37 7 4 4 13	60.5% 45.7% 8.6% 4.9% 4.9% 16.0%
`CHANGE_NETWORK_STATE`		✔ Turn your WiFi on or off ✗ Send information to the application's server	36 13	52.9% 19.1%

How to ask for permission? (Felt et al.,)



PRO

Applicable to all permissions, even advance approval

CON

Interruptive, looks like EULAs, habit-forming

INSTALL-TIME WARNINGS

How to ask for permission? (Felt et al., 2012)

PRO

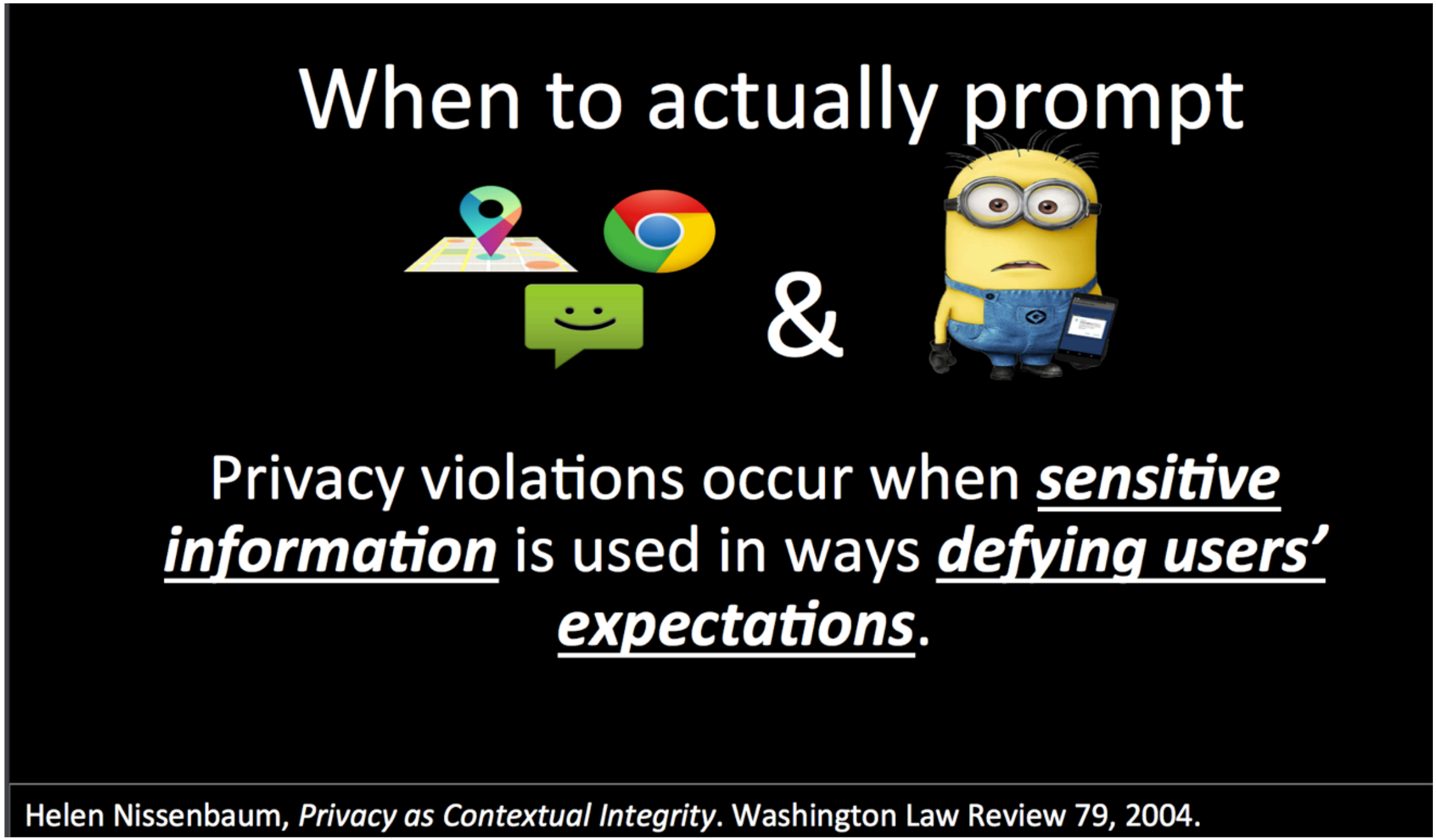
Applicable to almost all permissions

CON

Interruptive, habit-forming, not useful for advance approval

RUNTIME CONSENT DIALOGS

Android Permissions Remystified: A field study of Contextual Integrity (Wijesekera et al. 2015)



Android Permissions Remystified: A field study of Contextual Integrity (Wijesekera et al.)

The experiment

36 Android smartphone users

6,048 hours of real-world use

27 million permission requests

Android Permissions Remystified: A field study of Contextual Integrity (Wijesekera et al.)

Users want a choice

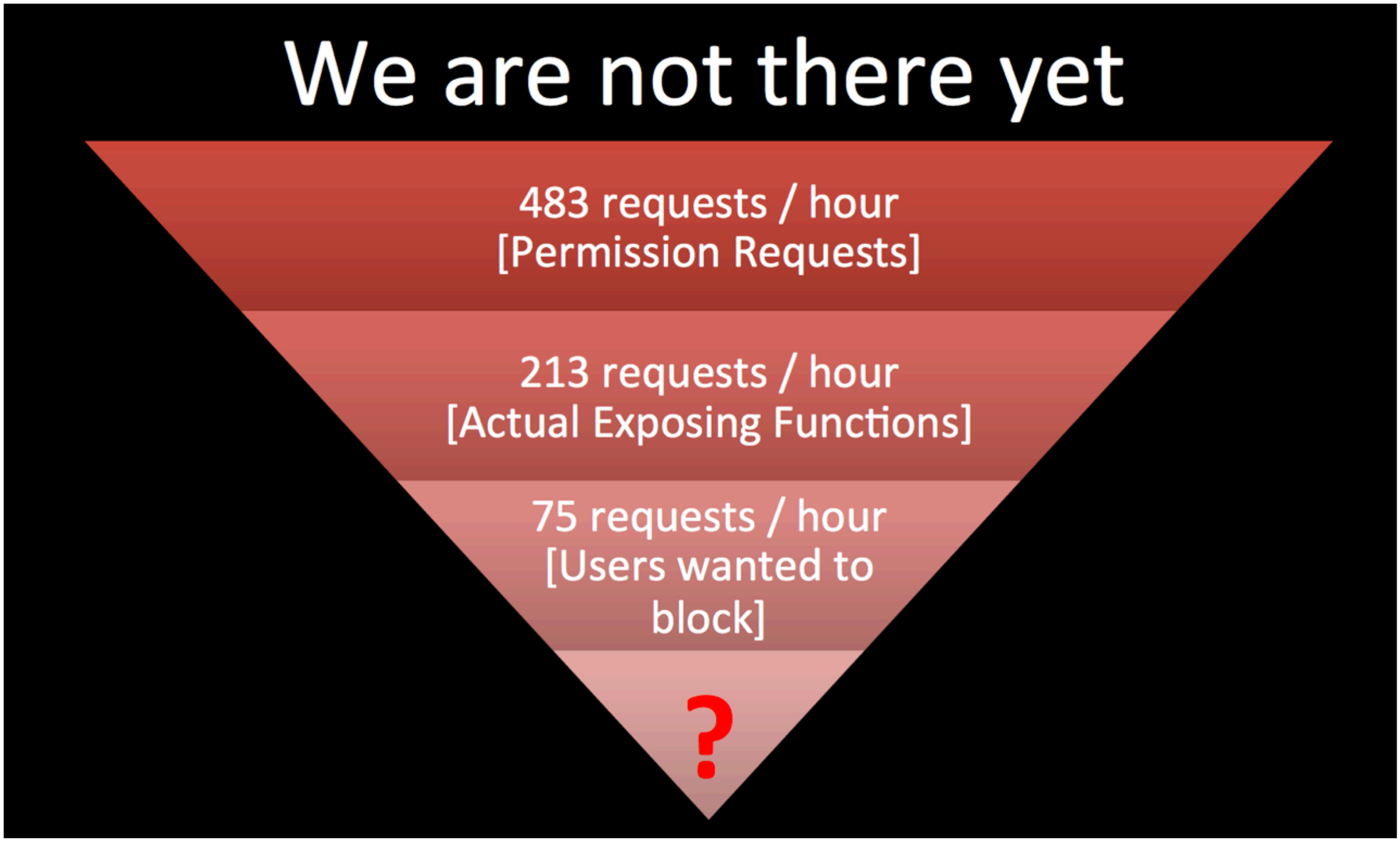
80% of users

would block at least one permission request.

35% of all requests

were deemed inappropriate.

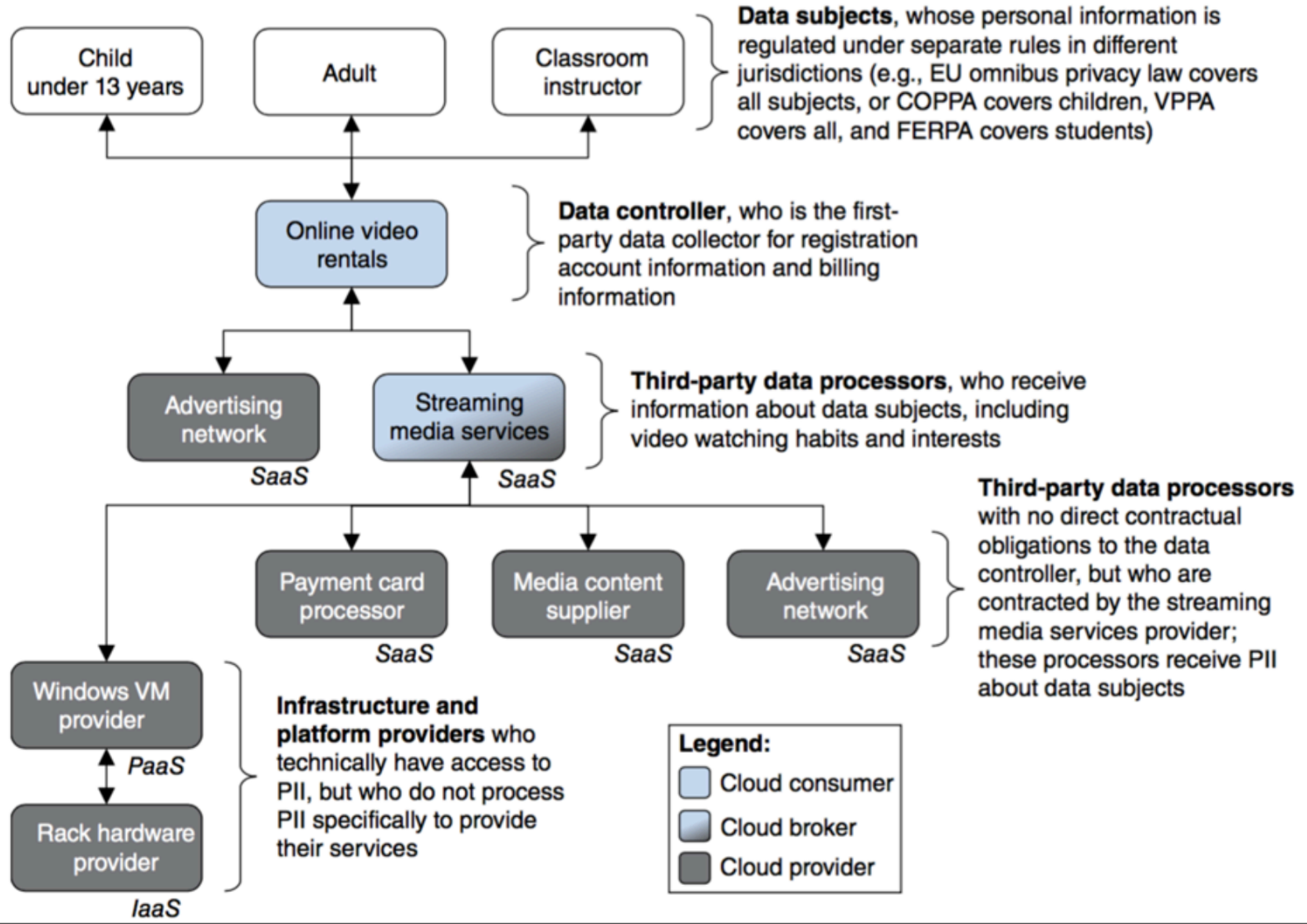
Android Permissions Remystified: A field study of Contextual Integrity (Wijesekera et al.)



what about the cloud?

“ensuring privacy in clouds”
Breaux and Pearson, 2016)

who is responsible for privacy?



“ensuring privacy in clouds”

who is responsible for privacy?

Lack of transparency, assurance, accountability

lack of clear responsibility

lack of trust

regulatory challenges

Service Level Agreements:

where is the data geographically?
(jurisdiction: which government will knock on your door/eavesdrop you?)

what’s the scope of third party access?

what security practices are used?

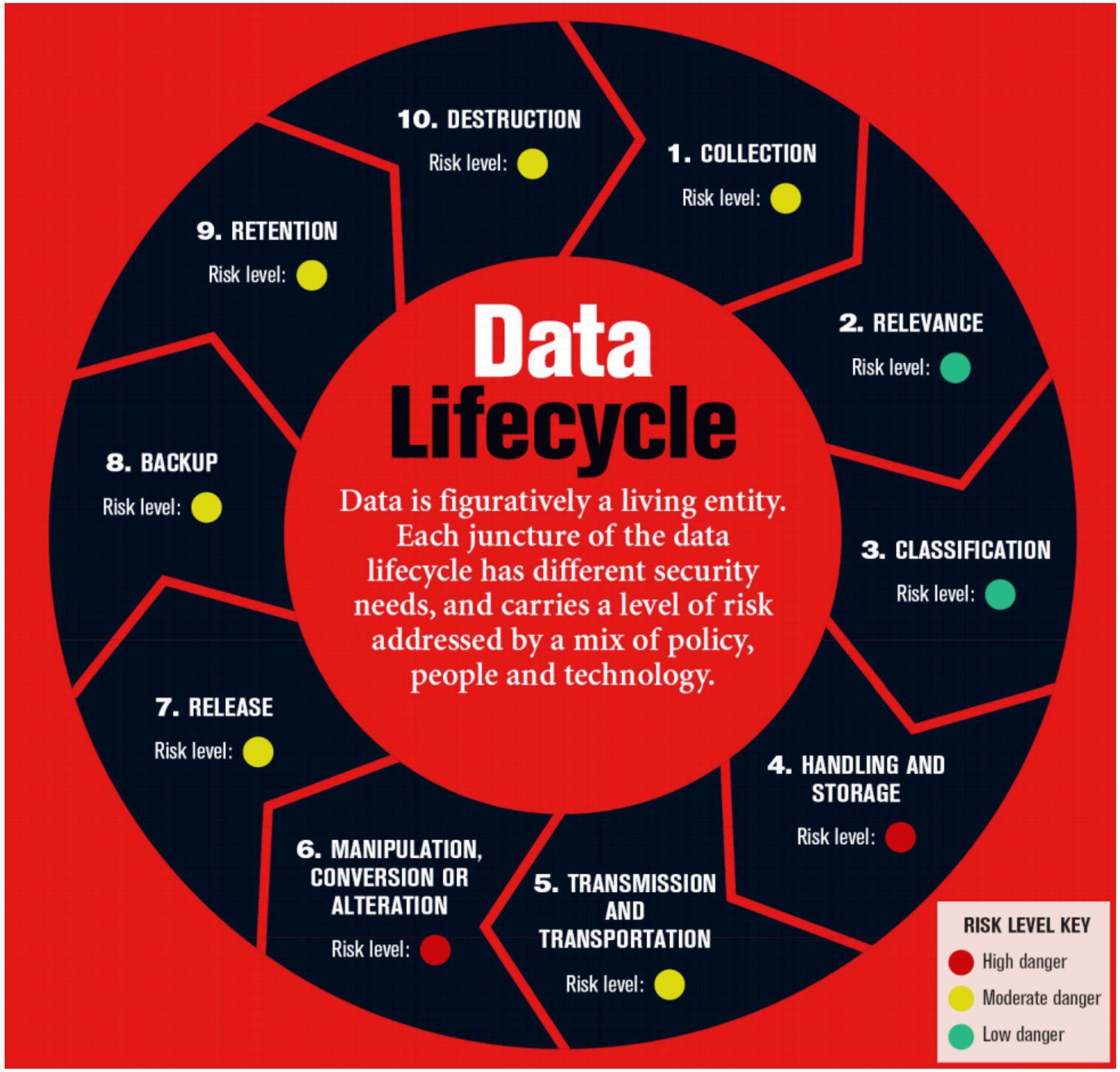
how are backups and data retention managed?

how is individual consent and subject access managed?

Service Level Agreements: users are dependent on data controller
(no leverage on contracts)

how are shifts in software engineering and the ecosystem relevant to privacy practice?

Data Protection Laws are Data Centric



http://cdn.ttgtmedia.com/informationsecurity/images/vol4iss7/ism_v4i7_f4_DataLifecycle.pdf

Eddy

a privacy specification language

Return to Main Page

Eddy is a privacy requirements specification language that privacy analysts can use to express requirements over acts to collect, use, transfer and retain personal and technical information. The language uses a simple SQL-like syntax to express whether an action is permitted or prohibited, and to restrict those statements to particular data subjects and purposes. The Eddy specifications are compiled into Description Logic to automatically detect conflicting requirements and to trace data flows within and across specifications. Each specification can describe an organization's data practices, or the data practices of specific components in a software architecure.

For further technical details on Eddy, please see our relevant publications:

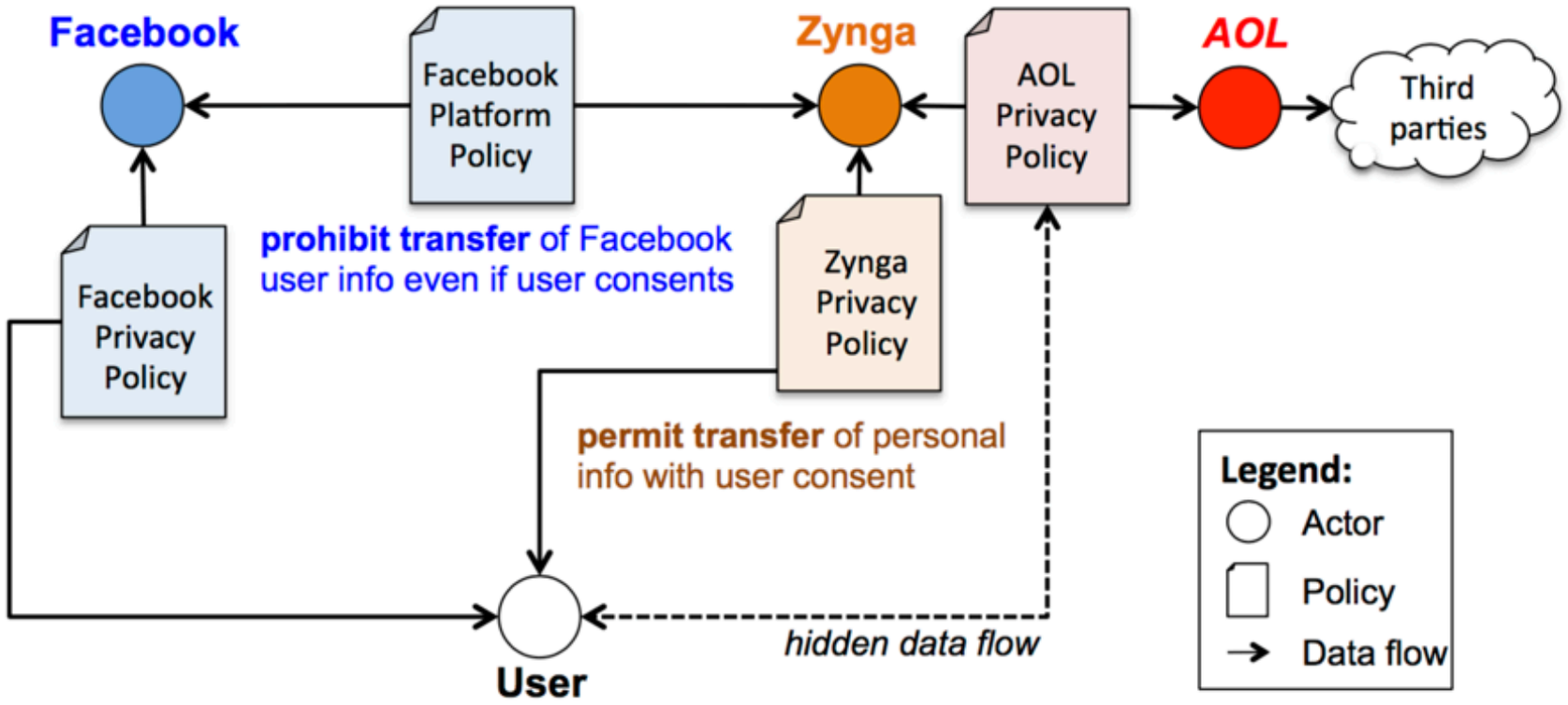
1. **Detecting Repurposing and Over-collection in Multi-party Privacy Requirements Specifications**
   Travis D. Breaux, Daniel Smullen, Hanan Hibshi. To Appear: *23rd IEEE International Requirements Engineering Conference*, Ottawa, Canada, 2015. (pdf)

2. **Eddy, A Formal Language for Specifying and Analyzing Data Flow Specifications for Conflicting Privacy Requirements**
   Travis D. Breaux, Hanan Hibshi, Ashwini Rao. *Requirements Engineering Journal,* 19(3): 281-307, 2014. (doi). This an extended journal version of our conference paper (doi) that was nominated for best paper and presented at IEEE RE'13 (slides)

We provide interactive examples below to demonstrate the Eddy language, and the Java source code is available on GitHub (source) under GPLv2.

View and analyze an existing example
Example specification to illustrate conflict analysis
Example specification to illustrate flow analysis
Example specification to illustrate use limitation analysis

Privacy and data supply chain



Privacy policies contain privacy requirements for data that flow within a data supply chain; conflicts can exist among these requirements; repurposing can be an issue

Specifying privacy requirements

- Expressing Modality in Description Logic (DL)
  - Obligation $\sqsubseteq$ Permission
  - *Conflict* $\equiv$ *Permission* $\sqcap$ *Prohibition*
- Actions
  - Collect, Use and Transfer
- Actions have following DL Roles
  - hasObject.Datum – the object of the action (data element)
  - hasSource.Actor – the source of the object (an actor)
  - hasPurpose.Purpose – the purpose of the action
  - hasTarget.Actor – the recipient of the object (an actor)

T. Breaux, A. Antón, J. Doyle. "Semantic Parameterization: A Process for Modeling Domain Descriptions." ACM TOSEM, 18(2): 5, November 2008

Carnegie Mellon University

Results of extended evaluation

Policy	S	D	Modality			Action		
			P	O	R	C	U	T
Facebook	105	39	15	4	25	6	15	14
Zynga	195	64	58	1	8	22	8	15
AOL	74			0	4	12	15	10

Extracte

Modalit

Actions

The Facebook API policy has more prohibitions (denoted by "R"), because it serves to regulate platform plugins, such as the Zynga game Farmville. In contrast, Zynga reserves more rights (denoted by "P") regarding how they collect, use and transfer user information.

Privacy Engineering should also include methodologies, techniques and tools in the Software Development Lifecycle (not just data management)

CLICK ON A SDL PHASE OR PRACTICE BELOW TO LEARN MORE

1. TRAINING	2. REQUIREMENTS	3. DESIGN	4. IMPLEMENTATION	5. VERIFICATION	6. RELEASE	7. RESPONSE
1. Core Security Training	2. Establish Security Requirements	5. Establish Design Requirements	8. Use Approved Tools	11. Perform Dynamic Analysis	14. Create an Incident Response Plan	
	3. Create Quality Gates/Bug Bars	6. Perform Attack Surface Analysis/ Reduction	9. Deprecate Unsafe Functions	12. Perform Fuzz Testing	15. Conduct Final Security Review	17. Execute Incident Response Plan
	4. Perform Security and Privacy Risk Assessments	7. Use Threat Modelling	10. Perform Static Analysis	13. Conduct Attack Surface Review	16. Certify Release and Archive	

LINDDUN (Wuyts, Scandariato, Joosen)

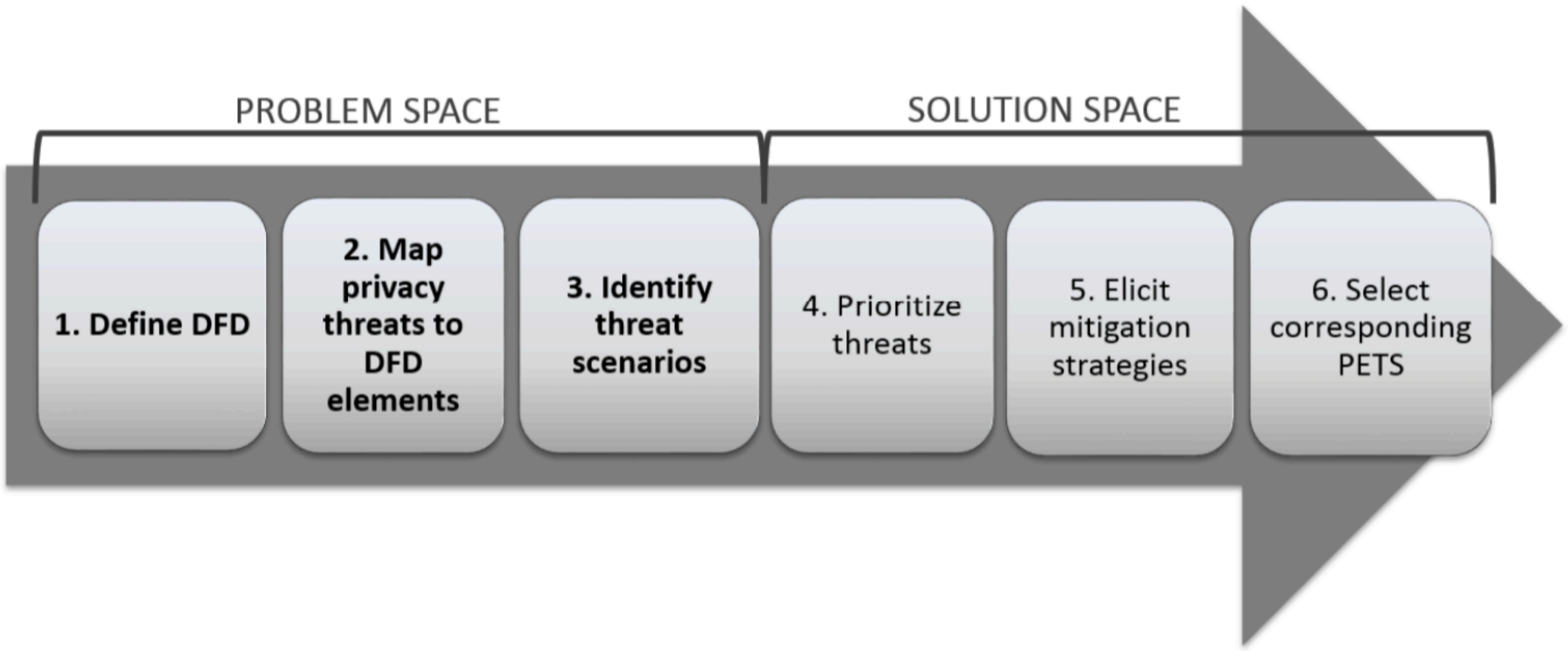
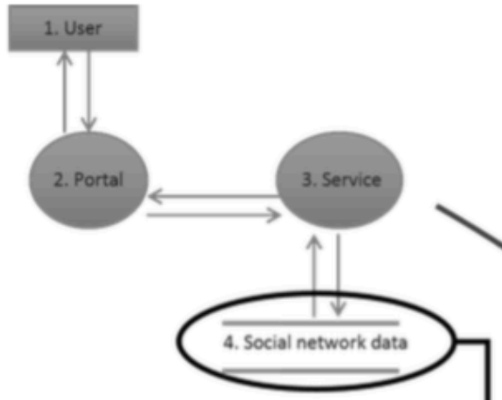


FIGURE 1: LINDDUN METHODOLOGY STEPS

https://distrinet.cs.kuleuven.be/software/linddun/index.php

1. Model DFD



2. Map threats to DFD

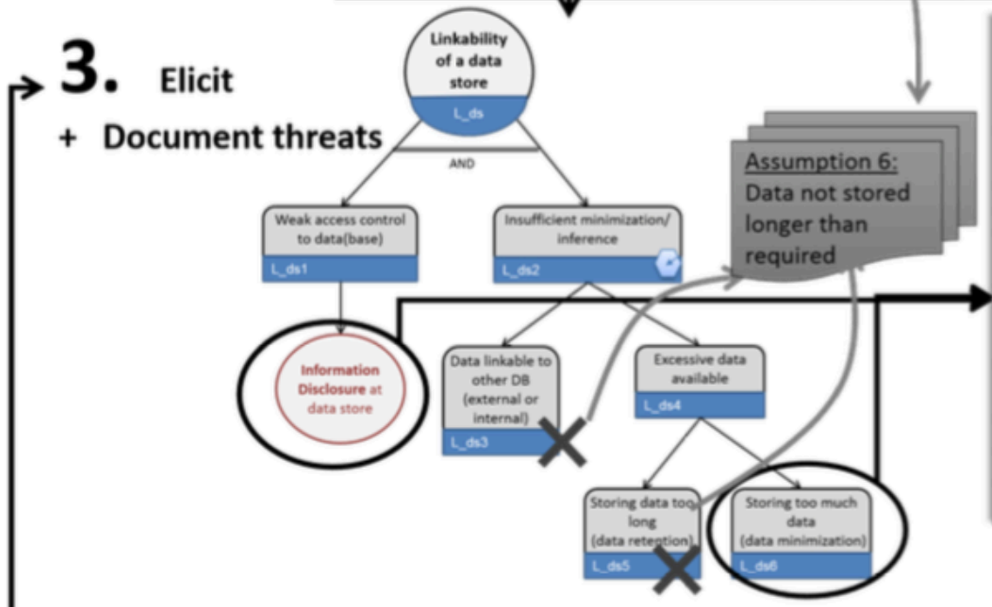
	Threat target	L	I	N	D	D	U	N
Data store	Social network db	X	X			X		X*
Data flow	User data stream (user-portal)	X	X			X		X*
	Service data stream (portal-service)							X*
	DB data stream (service – DB)							X*
Process	Portal							X*
	Social network service							X*
Entity	User	X	X				X	

assumptions

LINDDUN	L	I	N	D	D	U	N
Data store	X	X	X	X	X		X
Data flow	X	X	X	X	X		X
Process	X	X	X	X	X		X
Entity	X	X				X	

3. Elicit

+ Document threats



MUC 01: linking data

Summary: Data entries can be linked to the same person

Primary mis-actor: skilled insider / skilled outsider

Basic path:
bf1. The misactor gains access to the database
bf2. Because too much data are stored, information can be infered ...

Consequence: By combining the entries, the misactor has access to more information about the data subject than anticipated

Reference to threat tree node(s)*: L_ds1, L_ds6

Parent threat tree(s)*: L_ds

DFD element(s)*: 4. social network data

Remarks*: data not stored longer than required (assumpt. 6) + ...

mapping table completed ?

no

yes

Move to

LINDDUN *solution space*

Linkability (L) occurs when one can sufficiently distinguish whether 2 items of interest (IOI, such as requests from a user) are related

Identifiability (I) occurs when it is possible to pinpoint the identity of a subject (e.g., a user)

Non-repudiation (Nr) occurs when it is possible to gather evidence so that a party cannot deny having performed an action

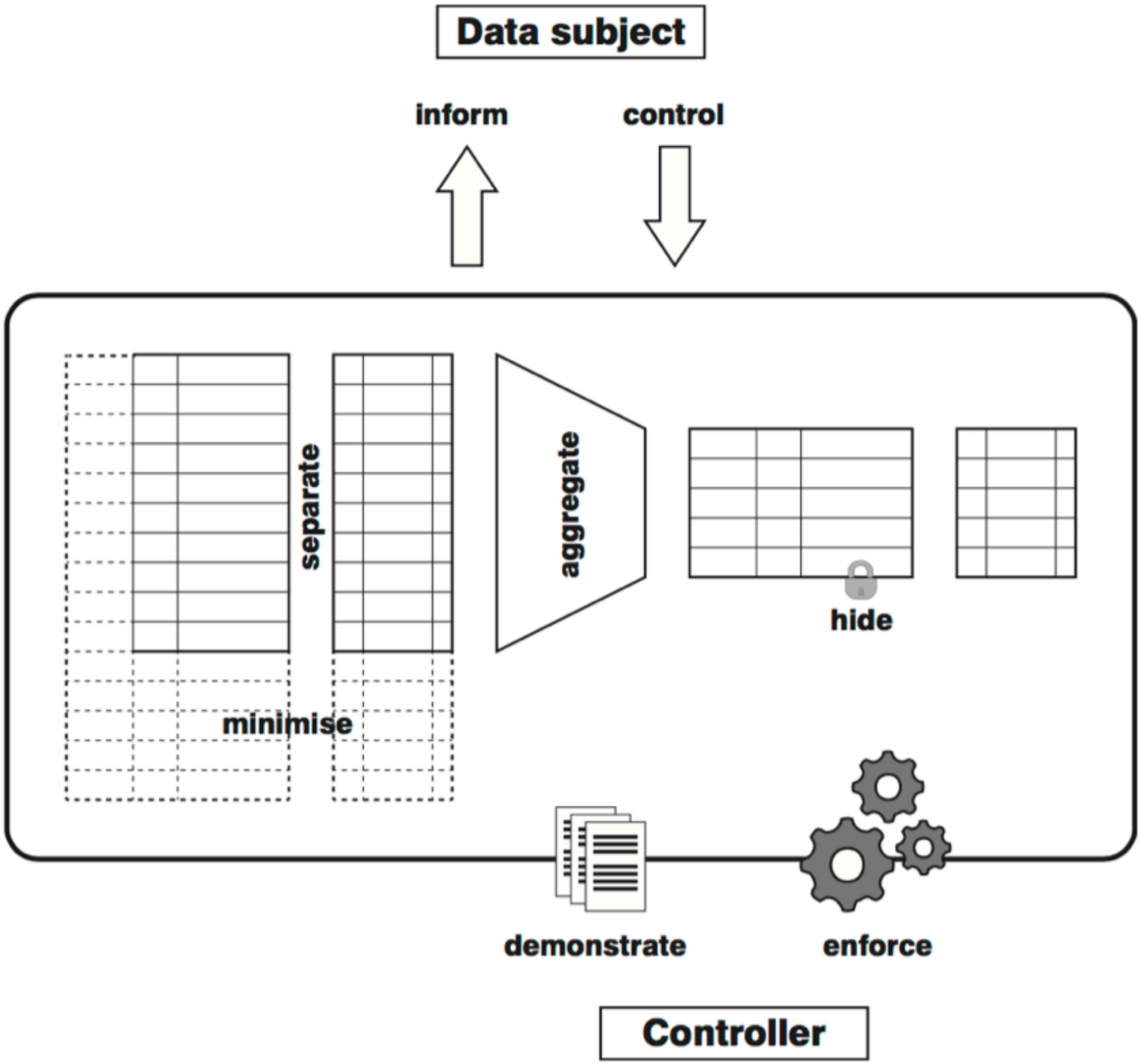
Detectability (D) occurs when one can sufficiently distinguish whether an IOI exists, e.g., in a system

Disclosure of information (Di) is the exposure of information to individuals who are not supposed to have access to it

Unawareness (U) occurs when the user is unaware of the information he is supplying to the system and the consequences of his/her act of sharing

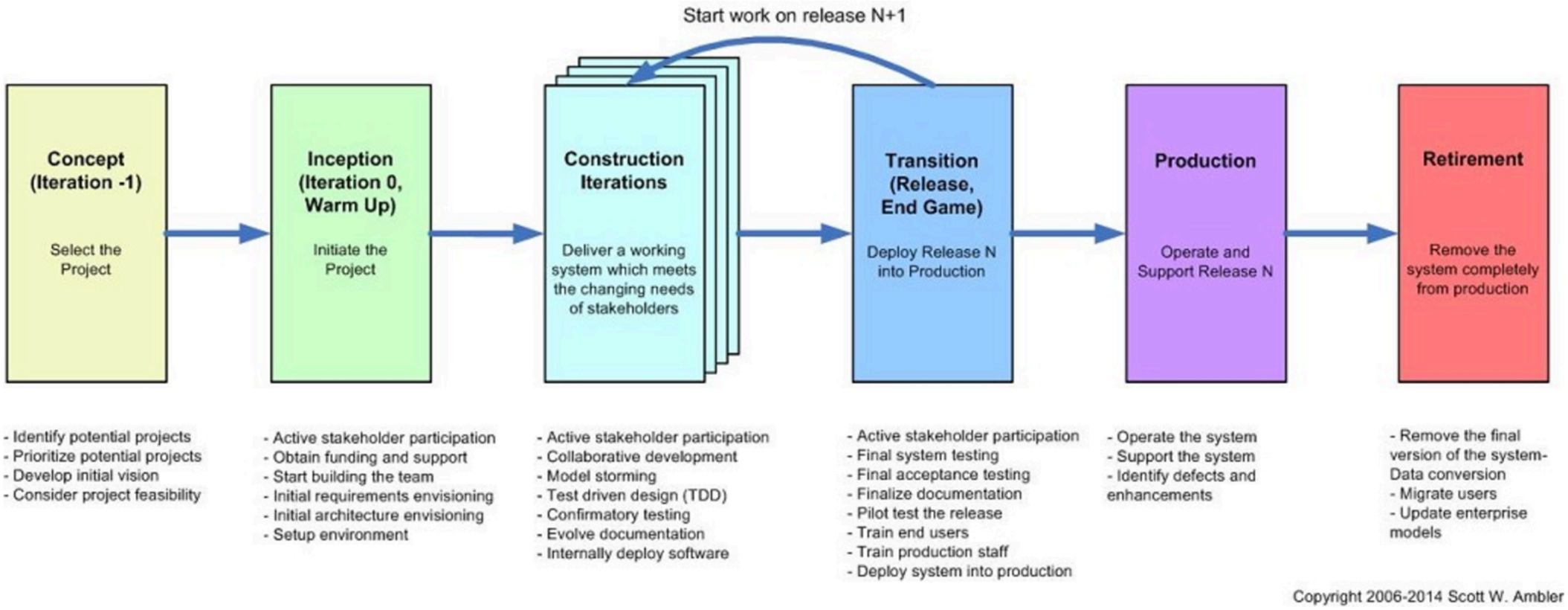
Non-compliance (Nc) occurs when the system is not compliant with the (data protection) legislation, its advertised policies and the existing user consents

Privacy Design Strategies (Hoepman et al.)

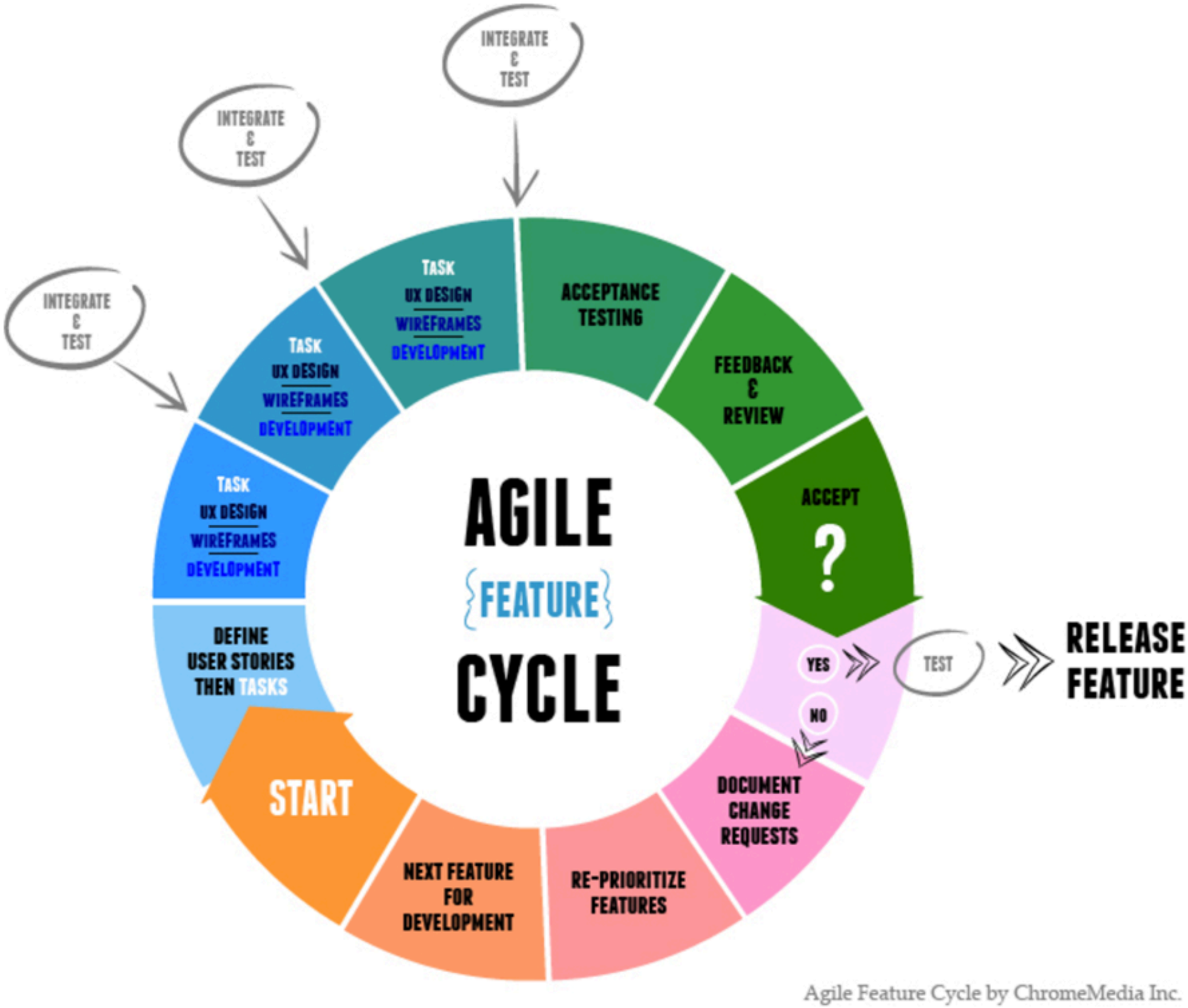


Agile Software Development Lifecycle is another animal

Figure 5. The Agile SDLC (high-level).



http://www.ambysoft.com/essays/agileLifecycle.html

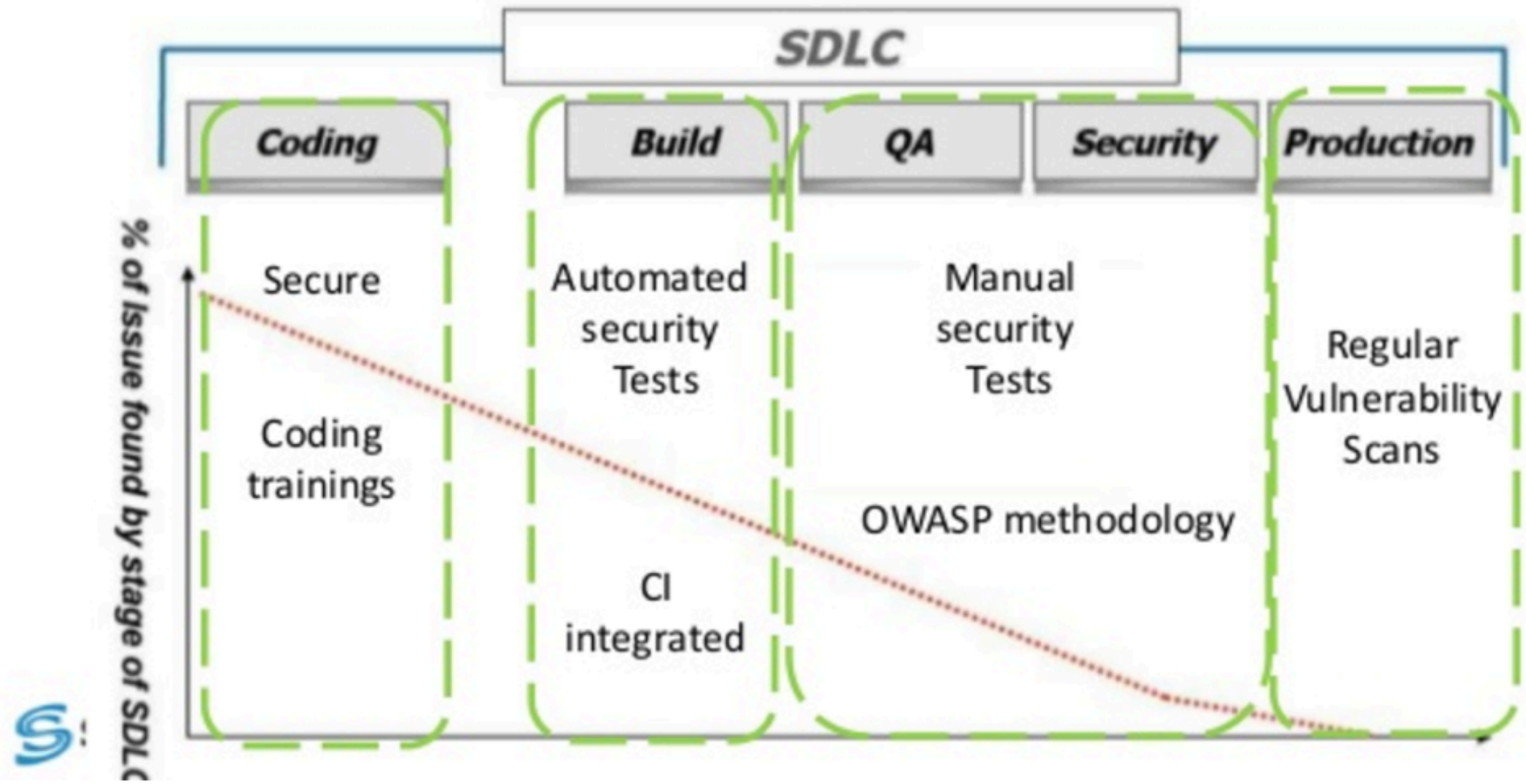


Agile Feature Cycle by ChromeMedia Inc.

http://chromemedia.com/how-we-work/

How it should look

With proper Security Program number of security defects should decrease from phase to phase



Slide deck: Agile and Secure SDLC by Nazar Tymoshyk, Cybersecurity Professional
https://www.slideshare.net/NazarTymoshyk/agile-and-secure-sflc, Published: October 2014.

what is the impact of

the turn to agile in software engineering practice

on computer science research in privacy?

OUTLOOK

- Privacy research will need to speak to existing SE approaches
  - domain specificity not enough: SE practices matter
- Future research: systemic empirical study of the agile turn and its impact on privacy research
  - evaluate the paradigmatic principles that guide privacy research
  - study feature inflation and its impact on privacy/security techniques
  - better understand behavioral analytics role in software engineering
  - the politics of new service metrics: opportunity to develop privacy metrics
- Investigate policy implications:
  - DP was developed during the time of mainframes!!!

references

- Many references in the slides. The agile turn based on:
- Philip E. Agre, Surveillance and capture: Two models of privacy, The Information Society, Vol. 10, Iss. 2, 1994
- Irina Kaldrack and Martina Leeker, There is no software, just services, Meson Press, 2015.
- Gürses and Van Hoboken, Privacy After the Agile Turn, Cambridge Handbook of Consumer Privacy, https://osf.io/27x3q/ (upcoming)
- For those who are interested:

  International Workshop on Privacy Engineering

  http://ieee-security.org/TC/SPW2017/IWPE/