# Privacy by Design

## Strategies & Patterns

**Jaap-Henk Hoepman**

*Digital Security (DS)*
*Radboud University Nijmegen, the Netherlands*
@xotoxot // ✉ *jhh@cs.ru.nl* // *www.cs.ru.nl/~jhh*



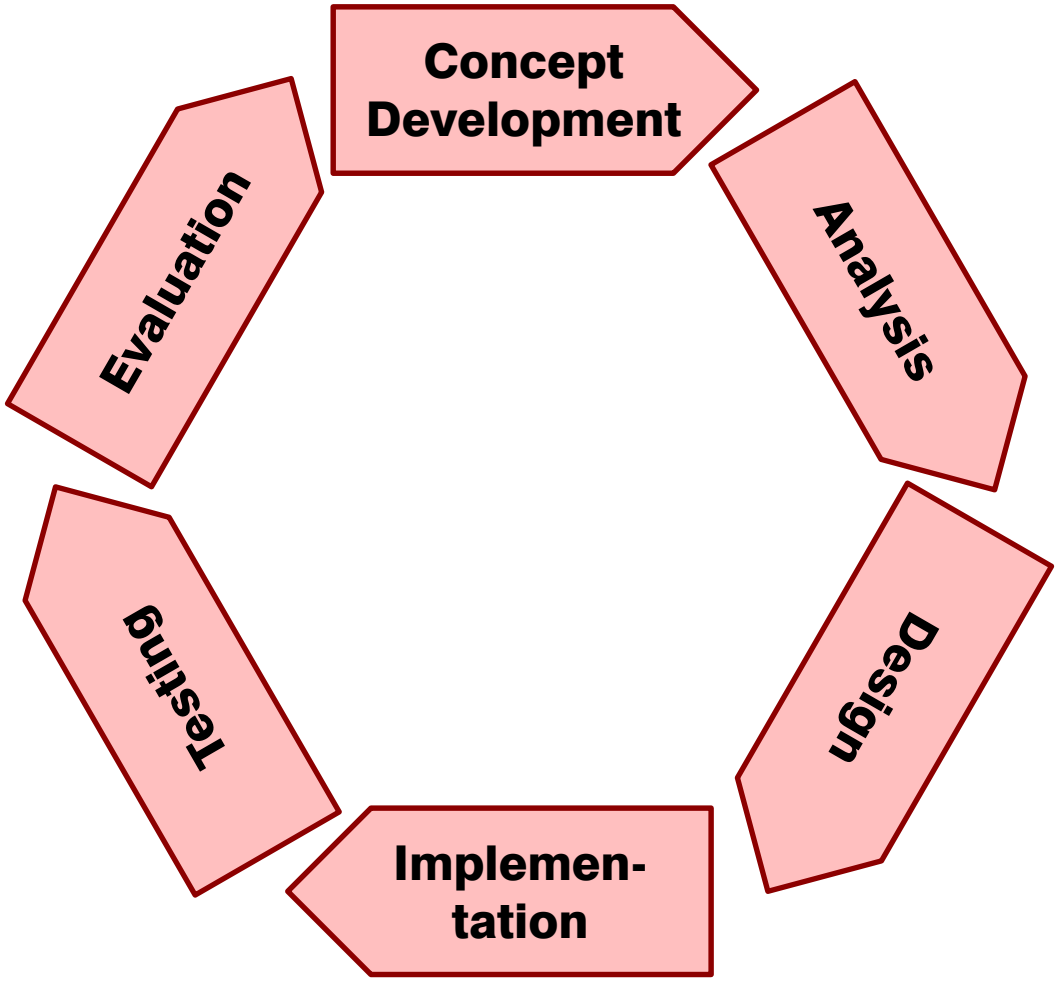
# Introduction

- Security
- Privacy
- Identity Management
- Internet of Things

Radboud University Nijmegen

Privacy & Identity Lab

# Software development cycle



# Privacy by design

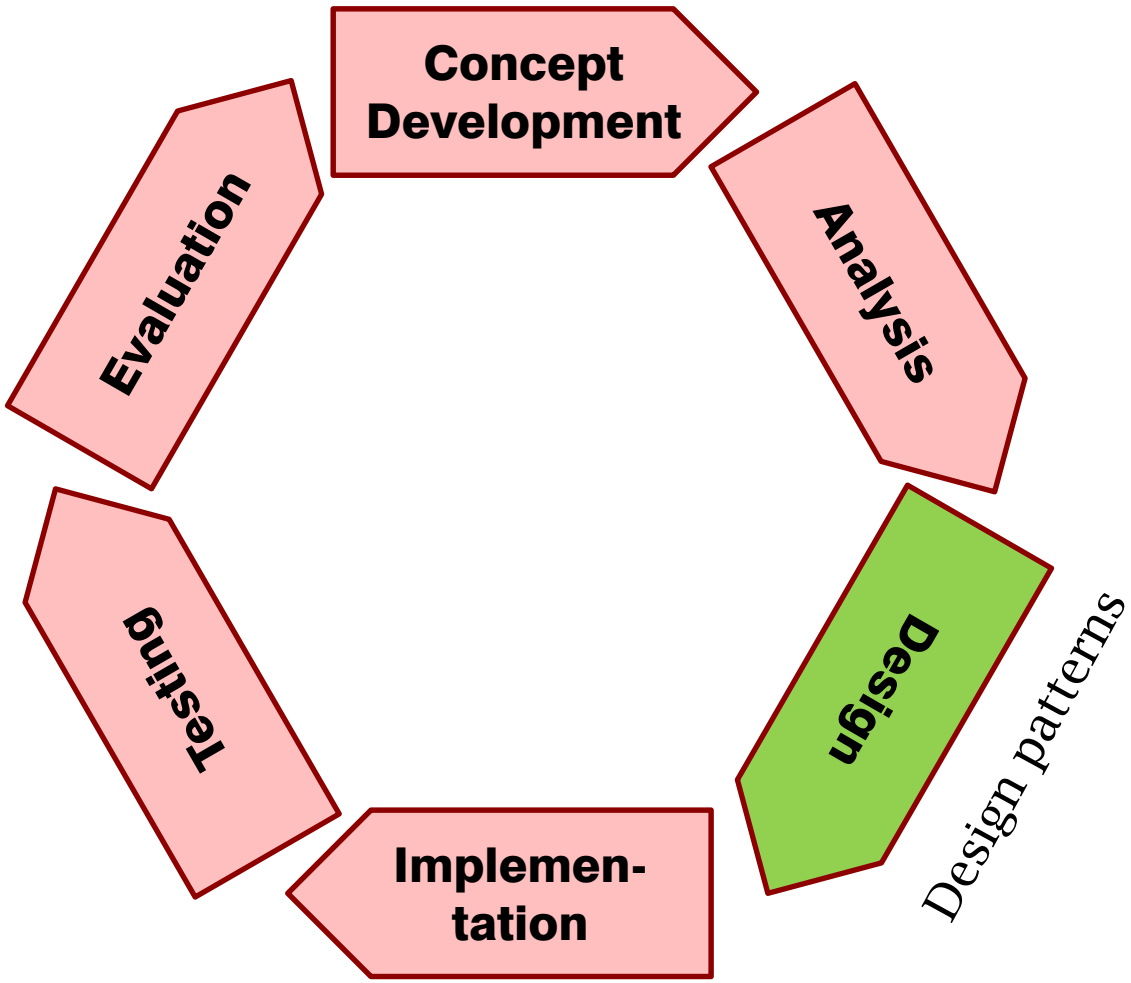
- **Protect privacy during technology development:**
  - From conception…
  - … to realisation.

**Through the full product development lifecycle**

# Levels of abstraction

- **Design pattern**
  - "Commonly recurring structure to solve a general design problem within a particular context"

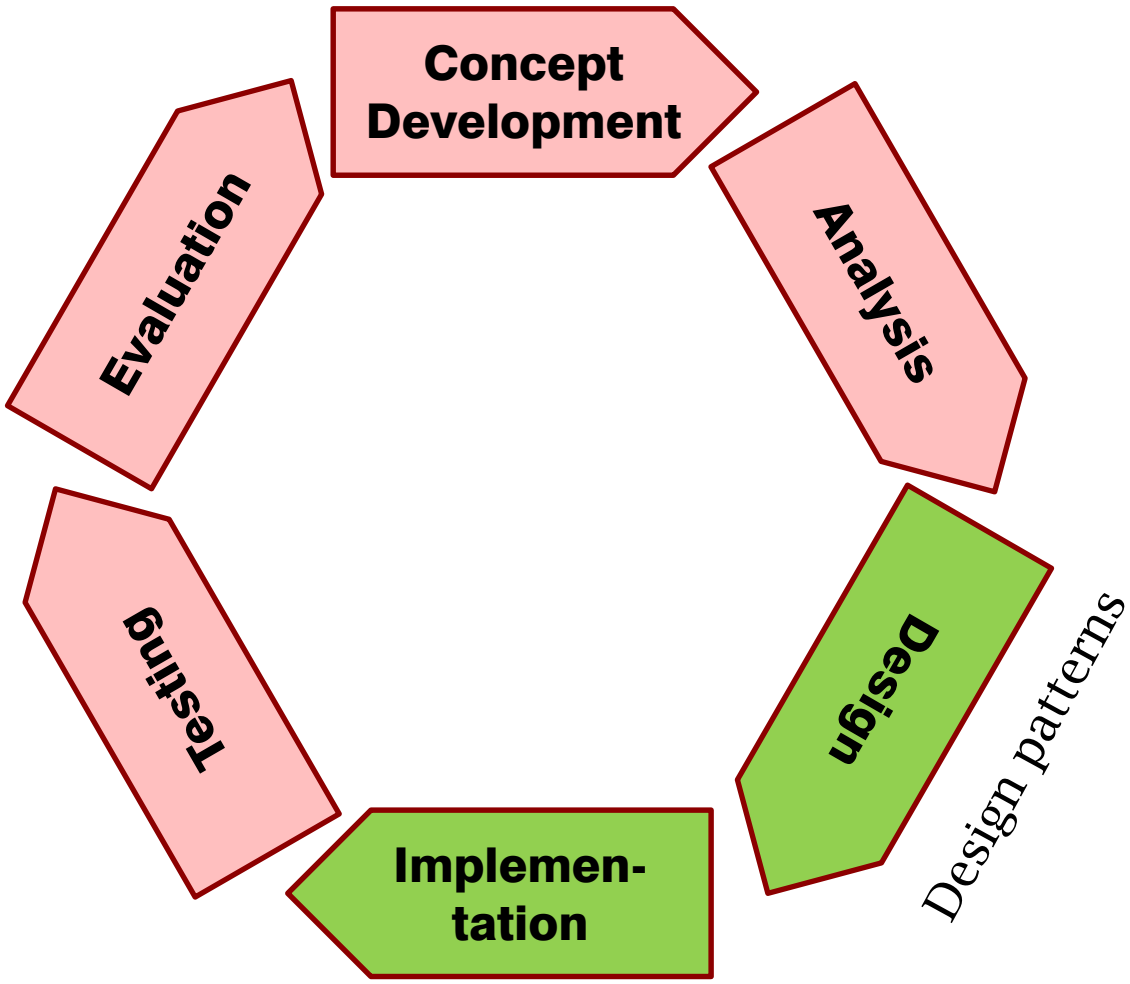
# Software development cycle



# Levels of abstraction

- **Design pattern**
  - "Commonly recurring structure to solve a general design problem within a particular context"
- **(Privacy enhancing) technology**
  - "A coherent set of ICT measures that protects privacy" – *implemented using concrete technology*

# Software development cycle

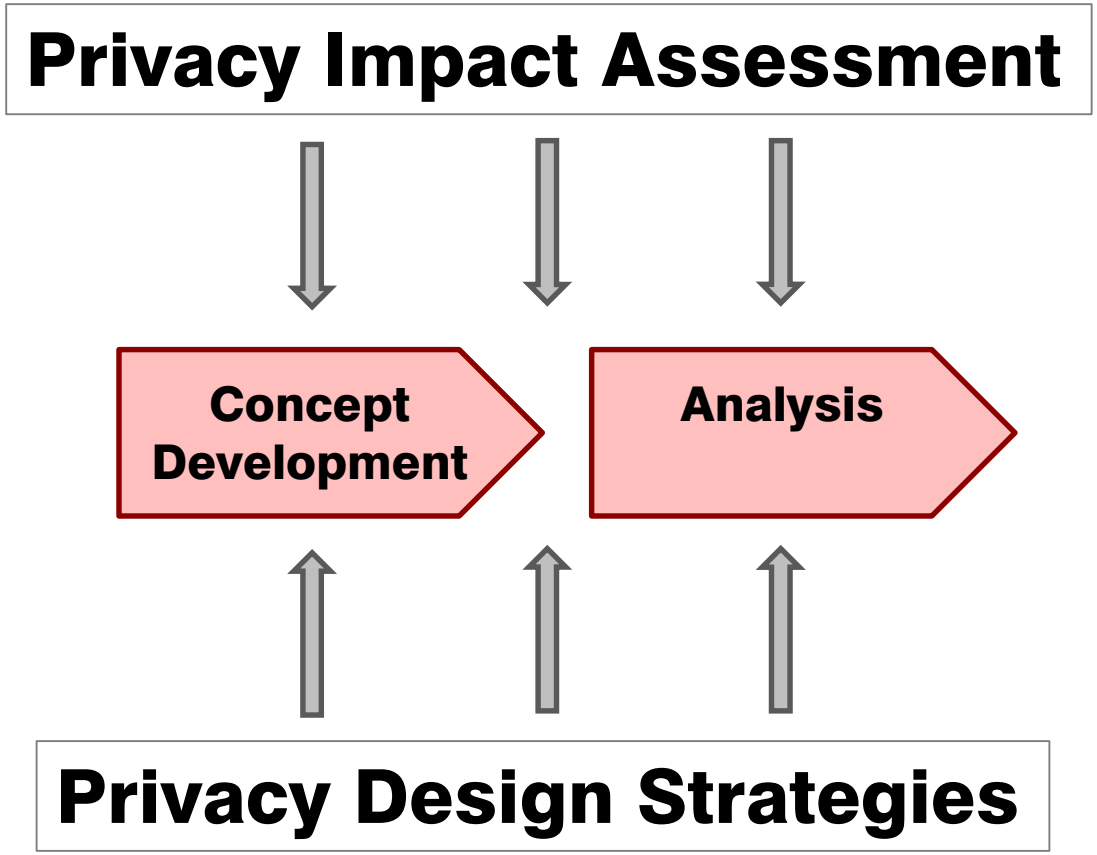



© 2004 www.planetside.co.uk

# Levels of abstraction

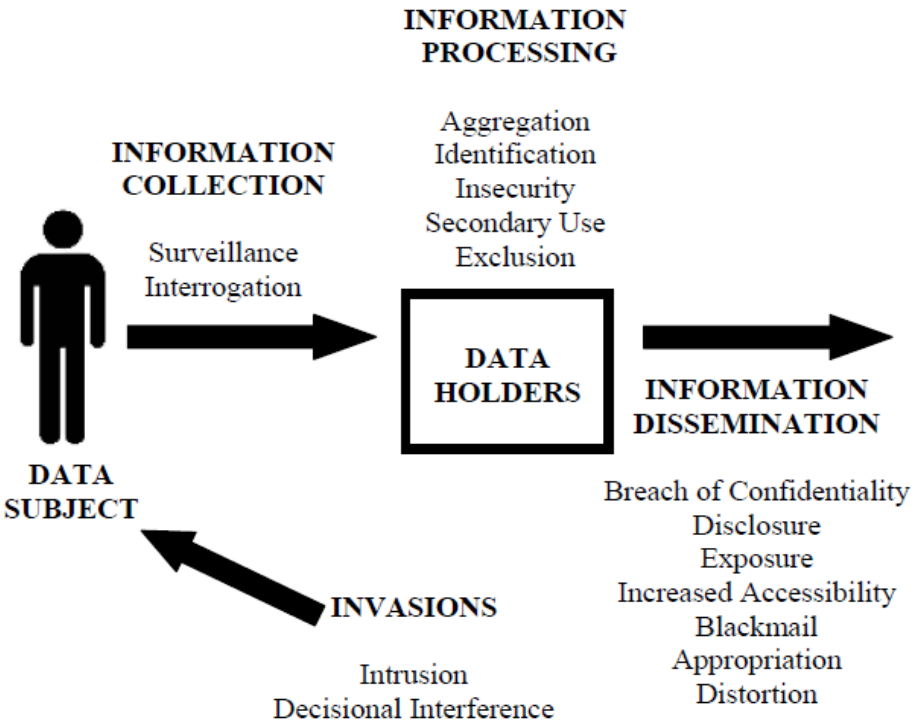
- **Design strategy**
  - "A basic method to achieve a particular design goal" – *that has certain properties that allow it to be distinguished from other basic design strategies*
- **Design pattern**
  - "Commonly recurring structure to solve a general design problem within a particular context"
- **(Privacy enhancing) technology**
  - "A coherent set of ICT measures that protects privacy" – *implemented using concrete technology*

# Concept development & analysis



# Eight Privacy Design Strategies

# Source #1: Solove




**Information storage**

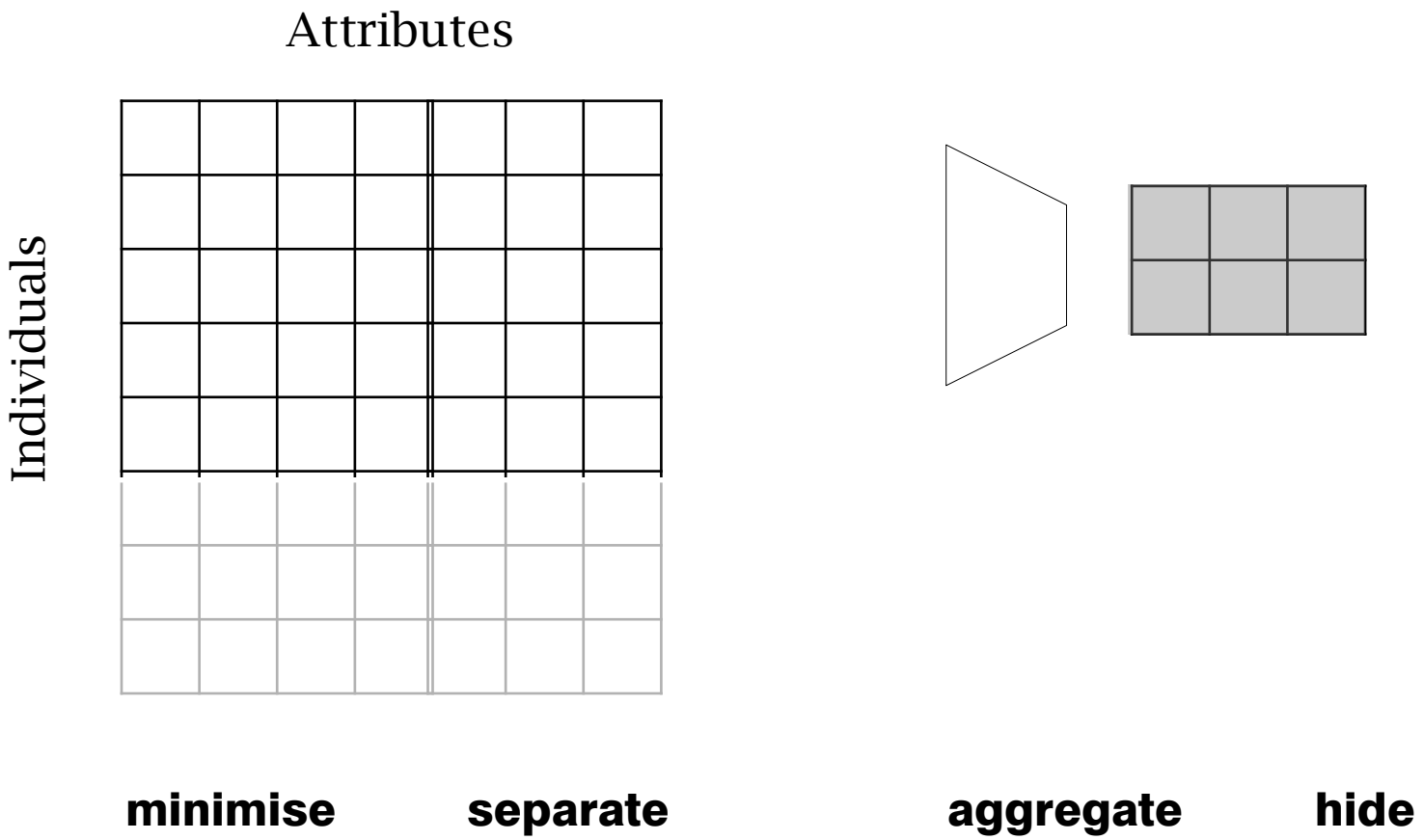
**Information flow**

# Source #2: data protection law

- **Core principles**
  - Data minimisation
  - Purpose limitation
  - Proportionality
  - Subsidiarity
  - Data subject rights: consent, (re)view
  - Adequate protection
  - (Provable) Compliance

**What happens if we want to apply these data protection principles to an information storage (ie database) system?**

# Database tables




**minimise** **separate** **aggregate** **hide**

# 8 privacy design strategies

- **Minimise**
  - The amount of PII should be minimal
- **Separate**
  - Process PII in a distributed fashion
- **Aggregate**
  - Process PII in the least possible detail
- **Hide**
  - PII should not be stored in plain view

# What did we cover

- **Core principles**
  - Data minimisation
  - Purpose limitation
  - Proportionality
  - Subsidiarity
  - Data subject rights: consent, (re)view
  - Adequate protection
  - (Provable) Compliance
- **Design strategies**
  - Minimise
  - Separate
  - Aggregate
  - Hide

# What did we cover

- **Core principles**
  - Data minimisation
  - Purpose limitation
  - Proportionality
  - Subsidiarity
  - Data subject rights: consent, (re)view
  - Adequate protection
  - (Provable) Compliance
- **Design strategies**
  - Minimise
  - Separate
  - Aggregate
  - Hide

# What did we cover

- **Core principles**
  - Data minimisation
  - Purpose limitation
  - Proportionality
  - Subsidiarity
  - Data subject rights: consent, (re)view
  - Adequate protection
  - (Provable) Compliance
- **Design strategies**
  - Minimise
  - Separate
  - Aggregate
  - Hide
  - Enforce
  - Inform
  - Control
  - Demonstrate

# 8 privacy design strategies

- **Minimise**
  - The amount of PII should be minimal
- **Separate**
  - Process PII in a distributed fashion
- **Aggregate**
  - Process PII in the least possible detail
- **Hide**
  - PII should not be stored in plain view
- **Enforce**
  - A privacy policy should be in place and be enforced
- **Inform**
  - Subjects should be informed when PII is processed
- **Control**
  - Subjects should have control over when/how PII is processed
- **Demonstrate**
  - Compliance to policies and legal requirements must be demonstrated

# What about design patterns?

| Strategy | Patterns | Coverage |
|---|---|---|
| **Minimise** | Select before you collect, anonymisation, …. | |
| **Separate** | Distribute, sector-specific pseudonyms | |
| **Aggregate** | Data fuzzing; coarse-grained location | |
| **Hide** | Encryption, onion routing, ….. | |
| **Enforce** | Access control, privacy licenses | |
| **Inform** | P3P (?) | |
| **Control** | Informed consent (?) | |
| **Demonstrate** | Privacy management system, logging | |

“Provides a scheme for refining the subsystems or components of a software system, or the relationships between them. It describes a commonly recurring structure of communicating components that solves a general design problem within a particular context.”

# (Privacy) Design Pattern Template

- **Name**
- **Intent**
- **Application context**
  - Including the problem it aims to solve
- **Implementation**
  - components & relationships
- **Consequences / Forces & Concerns**
  - Results, side-effects, trade offs
  - When to apply
  - When not to apply
- **Examples**
- **Related patterns**


- **Contribute**
  - http://wiki.science.ru.nl/privacy/Main_Page
- **See also: www.pilab.nl**