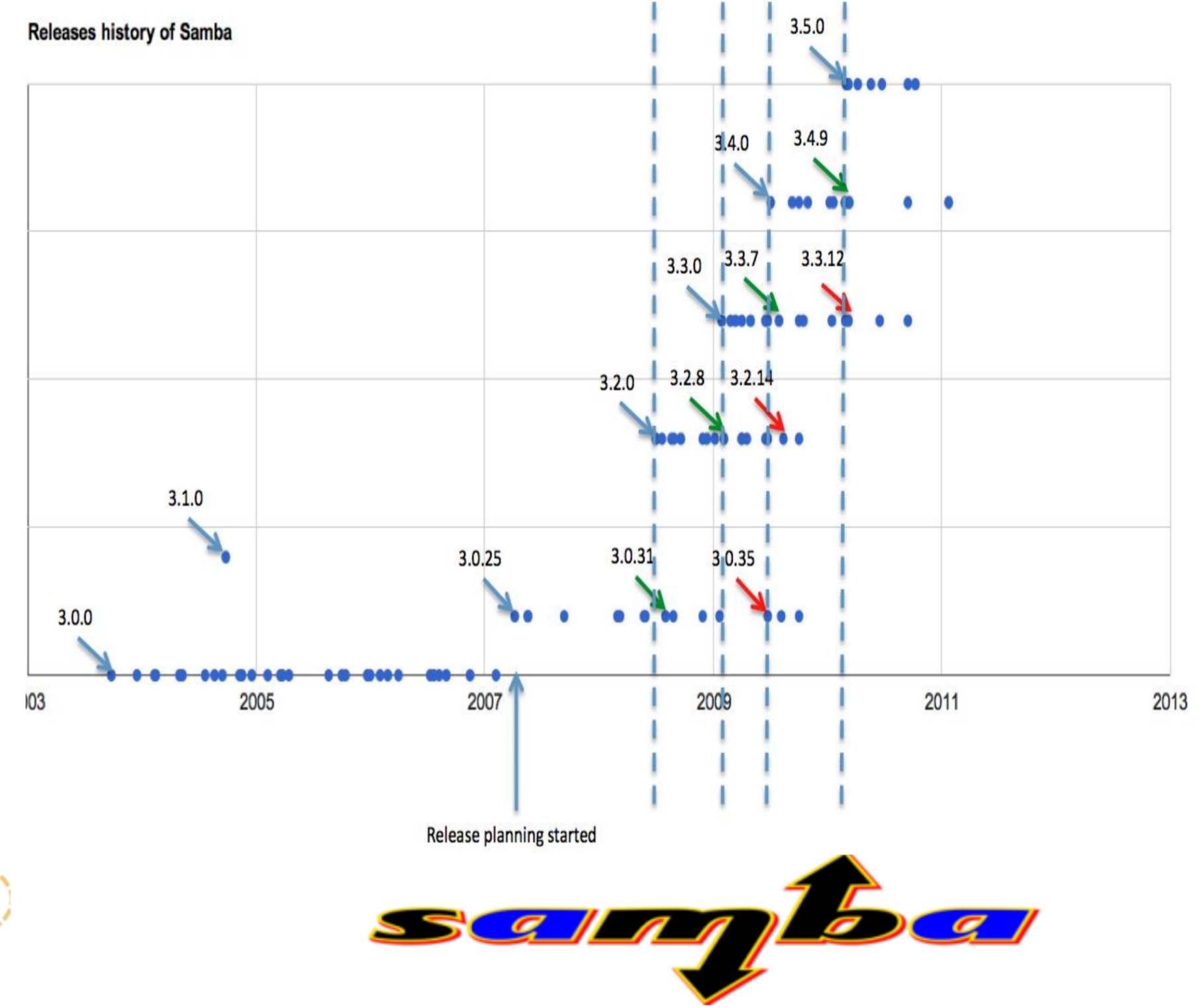
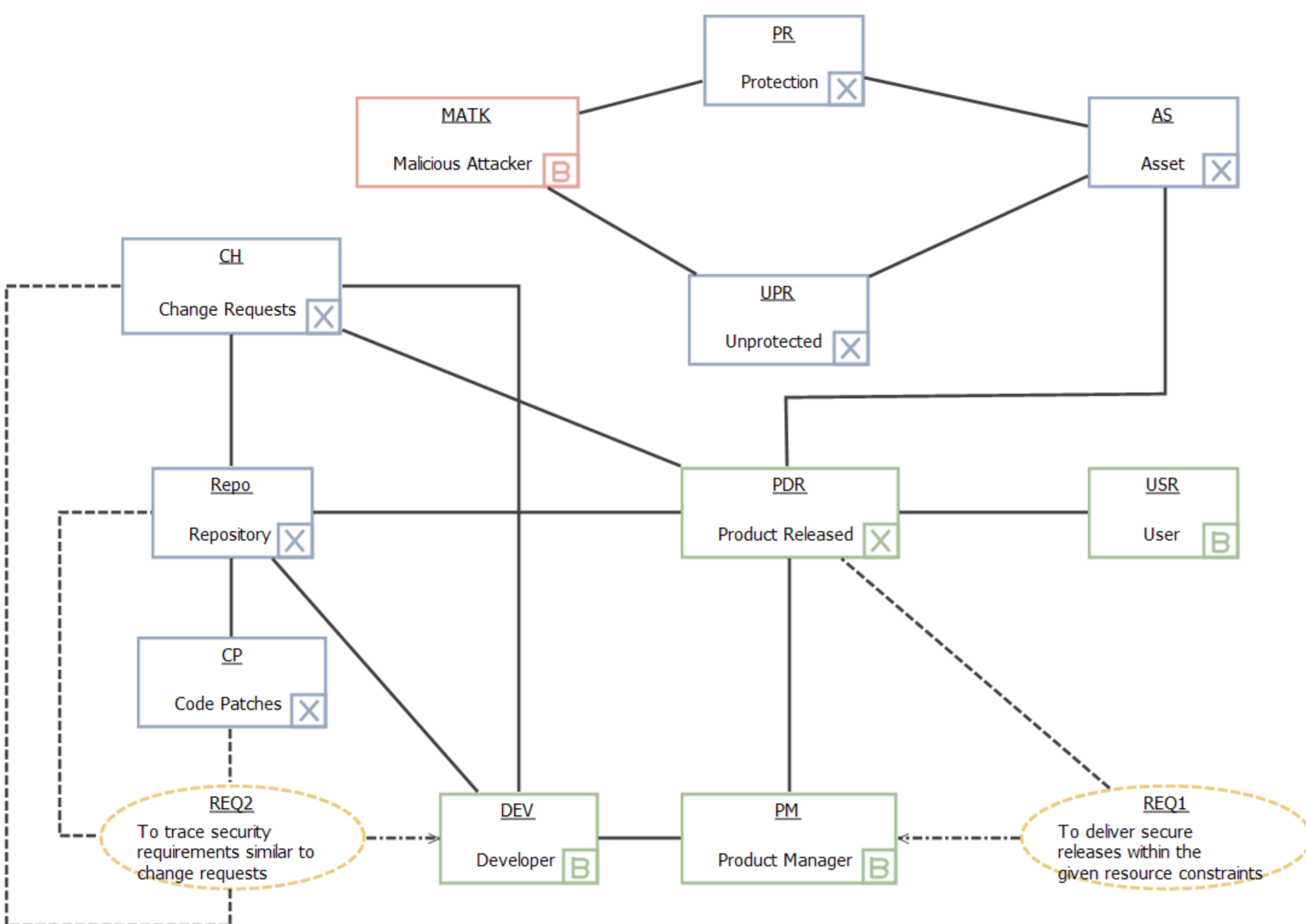


An Empirical Study of Planning Security Requirements for Evolving Software Systems

Saad Bin Saleem, Yijun Yu, Hui Yang, Bashar Nuseibeh, and Anne De Roeck

To deliver **secure releases**, software developers need to fix the system as soon as a change request concerns security, such as malicious attacks. To know which parts of the system are affected by the change request, it is useful to learn from the implementation of previous **security requirements** similar to the requested change. To release the product at the right time, **planning** of the changes is required to meet the constraints of given resources.



Traceability based learning

- Prepare **vocabularies** of “**asset**” and “**protection**”
- **Tokenise** the descriptions of both **bugs** and their **patches**
- **Learn** from **similarity** of **frequent terms** between current and past descriptions

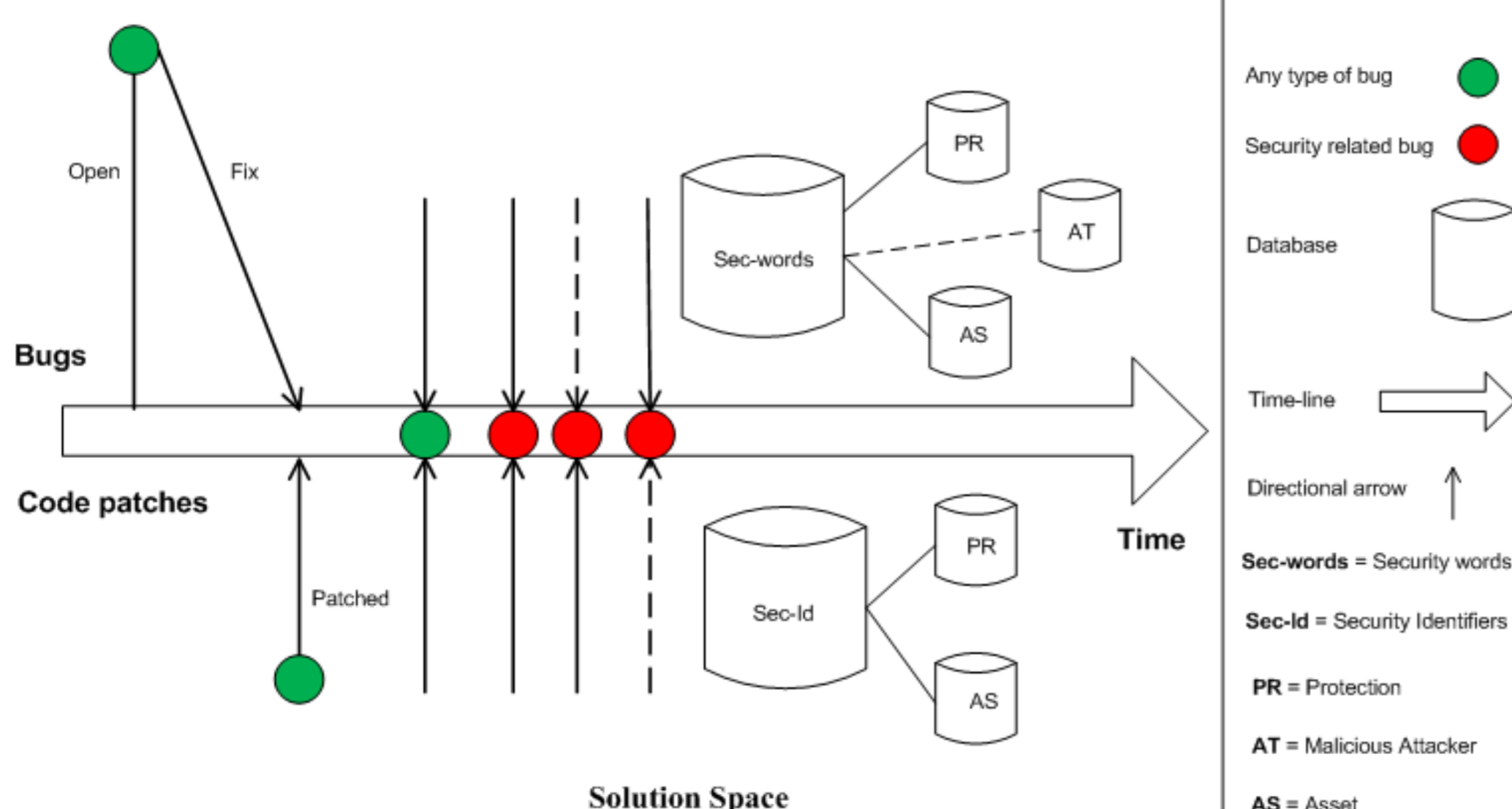
Secure release planning

- **Classify security and non-security** requirements
- Prepare inputs for the well known **release planner**™
- Generate **alternative plans** to deliver **secure product releases** at **right time**

Benefits

- Delivery of the **secure releases** at the **right time** based on the descriptions of change requests
- **Narrowing** the search scope for **developers** to trace the change requests to similar past security requirements

Problem Space



| Explanation | Alternative 1 | Alternative 2 | Alternative 3 | Alternative 4 | Alternative 5 |
|------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Degree of optimality | 99.2% | 99.2% | 96.7% | 94.3% | 94.3% |
| (Stakeholder feature points) | (35073) | (35039) | (34162) | (33340) | (33340) |
| Release | Alternative 1 | Alternative 2 | Alternative 3 | Alternative 4 | Alternative 5 |
| 3.0.25 | 5 | 4 | 5 | 5 | 5 |
| 3.3.0 | 13 | 13 | 13 | 12 | 12 |
| 3.4.0 | 2 | 2 | 2 | 2 | 2 |
| Total | 20 of 24 | 19 of 24 | 20 of 24 | 19 of 24 | 19 of 24 |