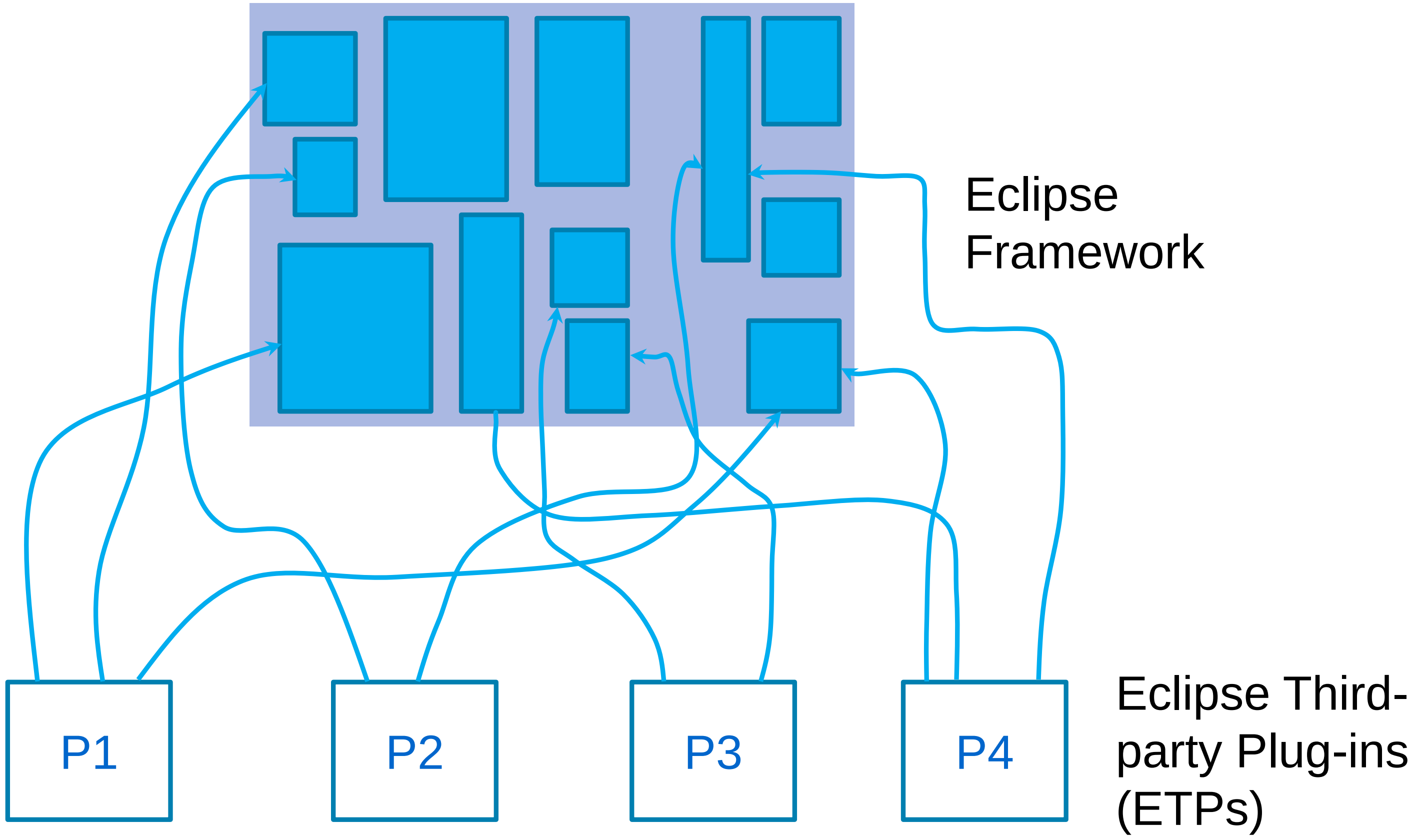
Compatibility Prediction of Eclipse Third-party Plug-ins in new Eclipse Releases

John Businge, Alexander Serebrenik, Mark van den Brand

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

The Eclipse Framework



Technische Universiteit
Eindhoven
University of Technology

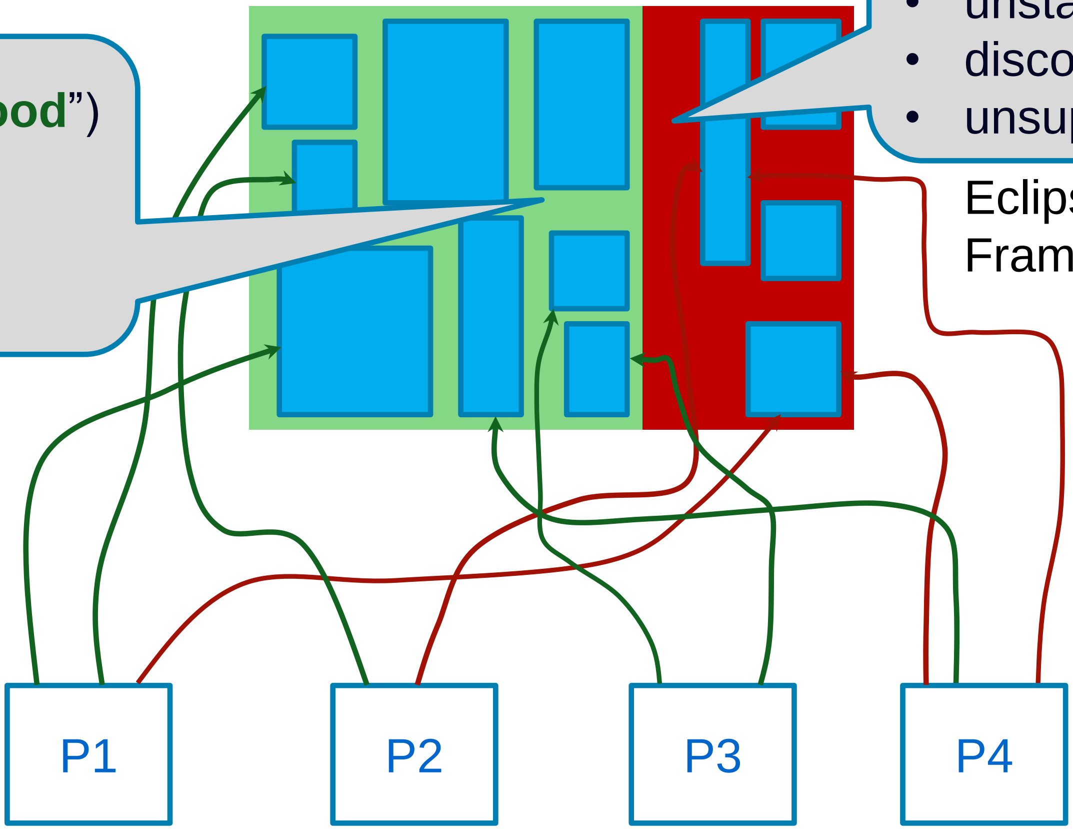
The Eclipse Framework …

Eclipse APIs(“**good**”)

- no “internal”
- stable,
- supported

Eclipse non-APIs (“**bad**”)

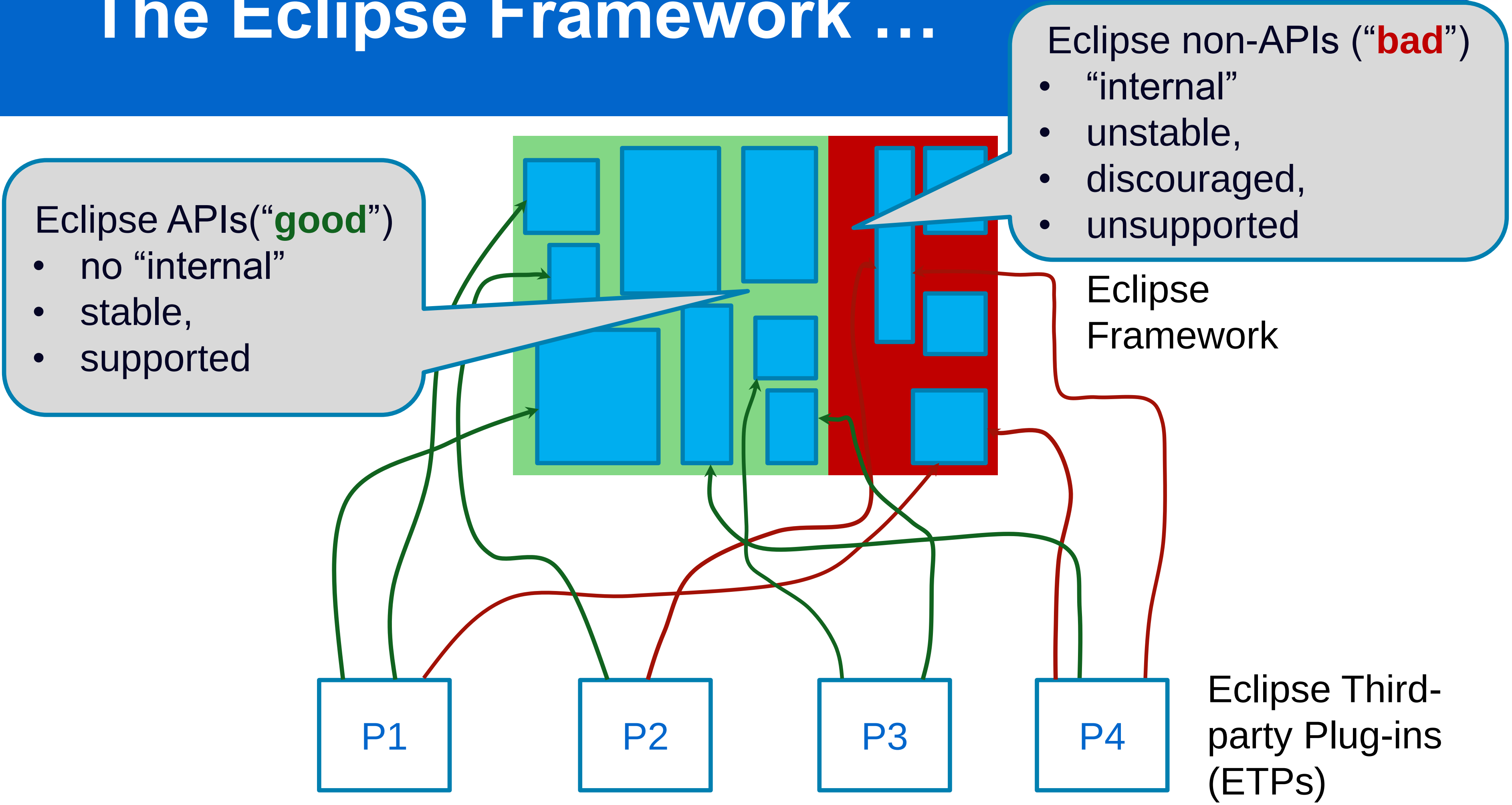
- “internal”
- unstable,
- discouraged,
- unsupported



Eclipse Framework

Eclipse Third-party Plug-ins (ETPs)

The Eclipse Framework …



Eclipse APIs(“**good**”)
- no “internal”
- stable,
- supported

Eclipse non-APIs (“**bad**”)
- “internal”
- unstable,
- discouraged,
- unsupported

- P3 – ETP-APIs
- P1, P2 and P4 – ETP-non-APIs

Motivation

- Previous study (Survival of ETPs – ICSM 2012), we tested compatibility of 345 ETP-APIs and 288 ETP-non-APIs with different Eclipse releases.
- Our observations:
  1. ETP-APIs always compatible in new Eclipse releases.
  2. Bad interfaces are the main cause of incompatibilities.
  3. Informally, found old bad interfaces stable.
- Formally verified observation 2.
- Trained prediction models
- Tested the prediction models.

Motivation

- Previous study (Survival of ETPs – ICSM 2012), we tested compatibility of 345 ETP-APIs and 288 ETP-non-APIs with different Eclipse releases.
- Our observations:
  1. ETP-APIs always compatible in new Eclipse releases.
  2. **Bad interfaces** are the main cause of incompatibilities.
  3. Informally, found old bad interfaces stable.
- Formally ~~verified~~ observation 2.
- Trained prediction models
- Tested the ~~prediction~~ models.

Compatibility prediction

- Requirements of compatibility prediction:
  - Current SDK compatible with ETP
  - Later SDK to make prediction
- We built 36 prediction models in total
- Models are bases on bad interfaces used by ETPs

ETP-non-APIs supported in Eclipse Releases

Eclipse	2.1	3.0	3.1	3.2	3.3	3.4	3.5	3.6	Total
# ETPs	29	48	34	40	38	36	33	30	288

Model Training

- Statistics – Binary Logistic Regression

$$\boldsymbol{P}(event) = 1/(1 + e^{-\boldsymbol{z}})$$

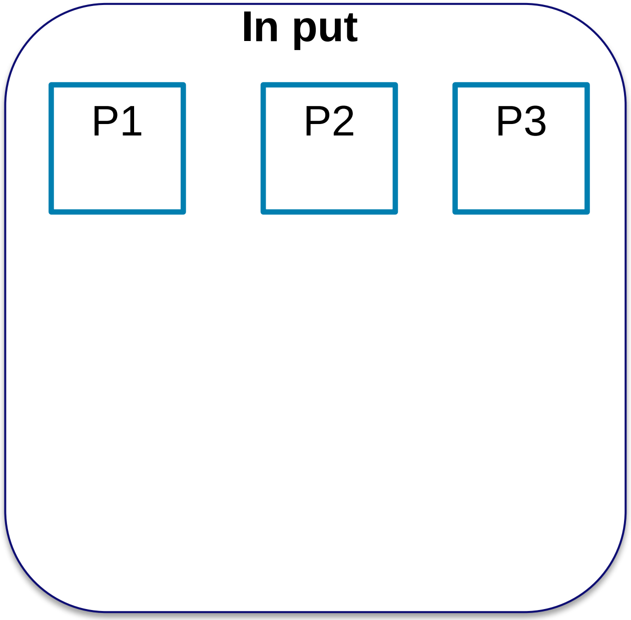
$$\boldsymbol{z} = b_0 + b_1X_1 + \ \dots + b_pX_p$$

$\boldsymbol{P}(event)$ –Dependent variable

p # Independent variables

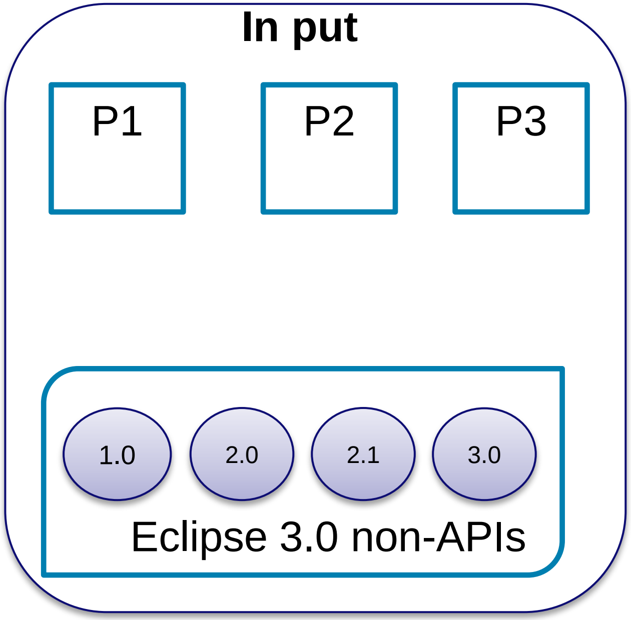
Model Training Example

ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0



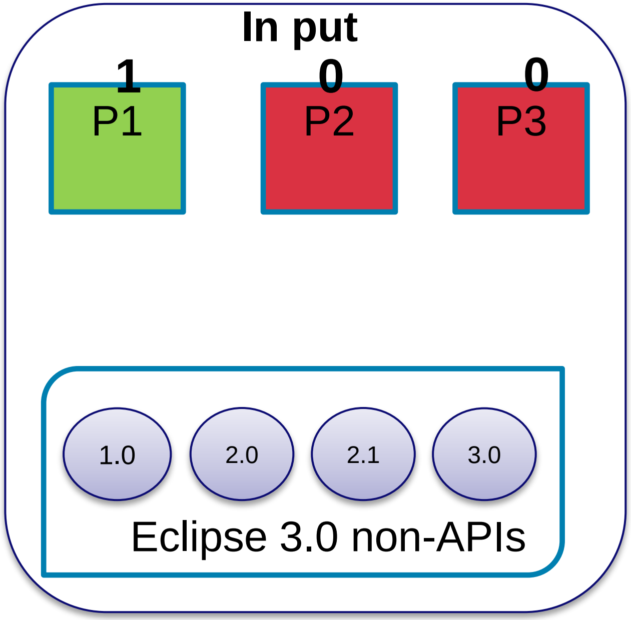
Model Training Example

ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0



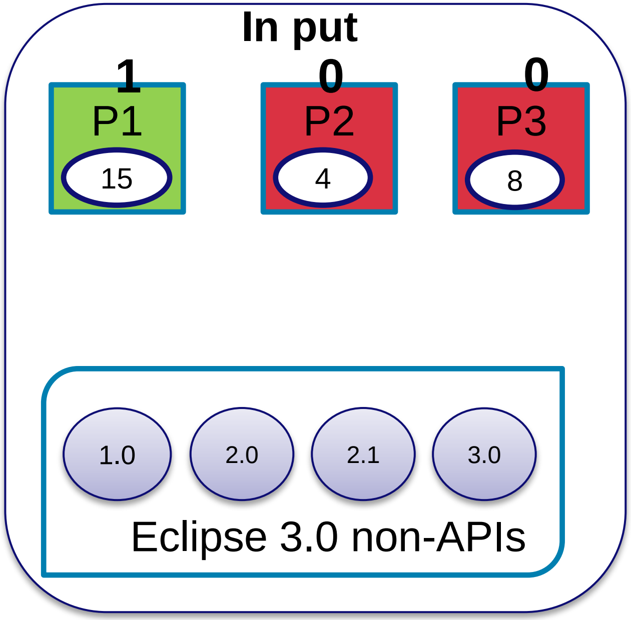
Model Training Example

ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0



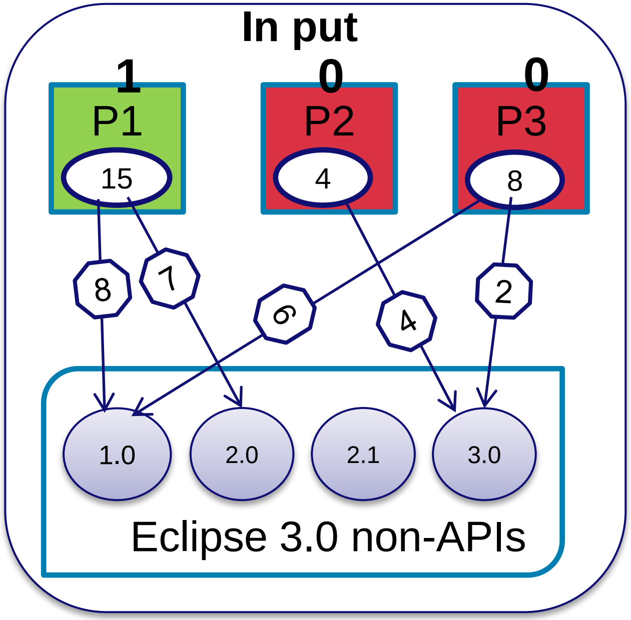
Model Training Example

ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0



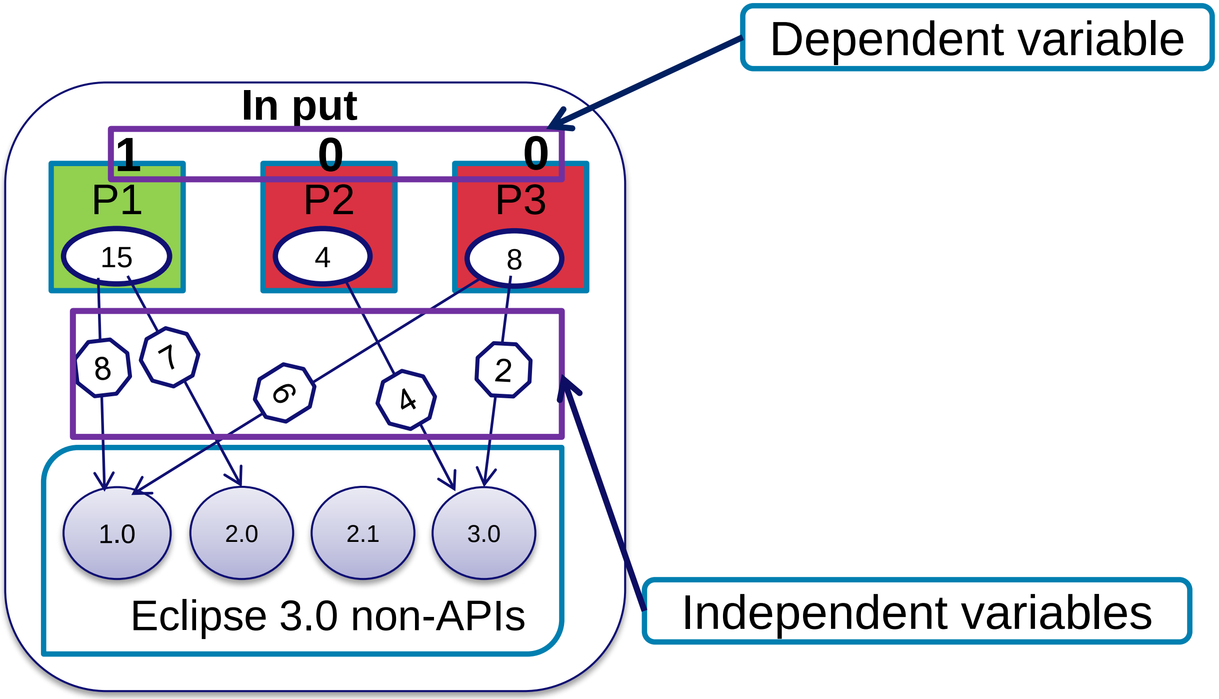
Model Training Example

ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0



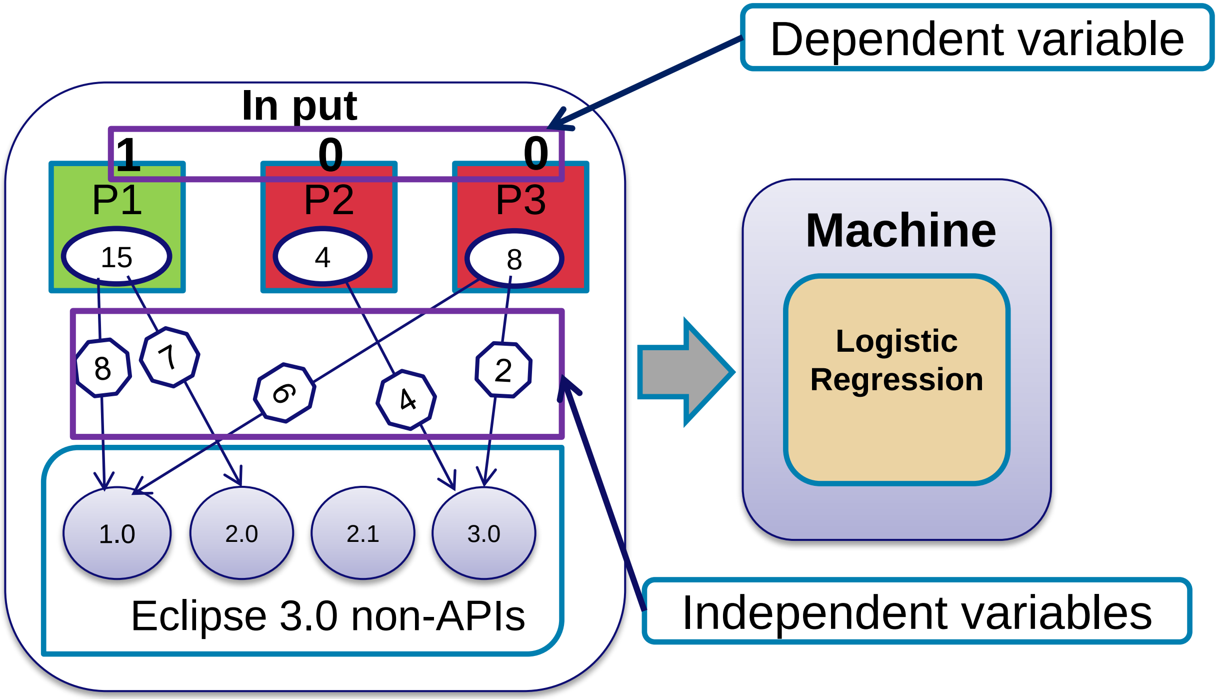
Model Training Example

ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0



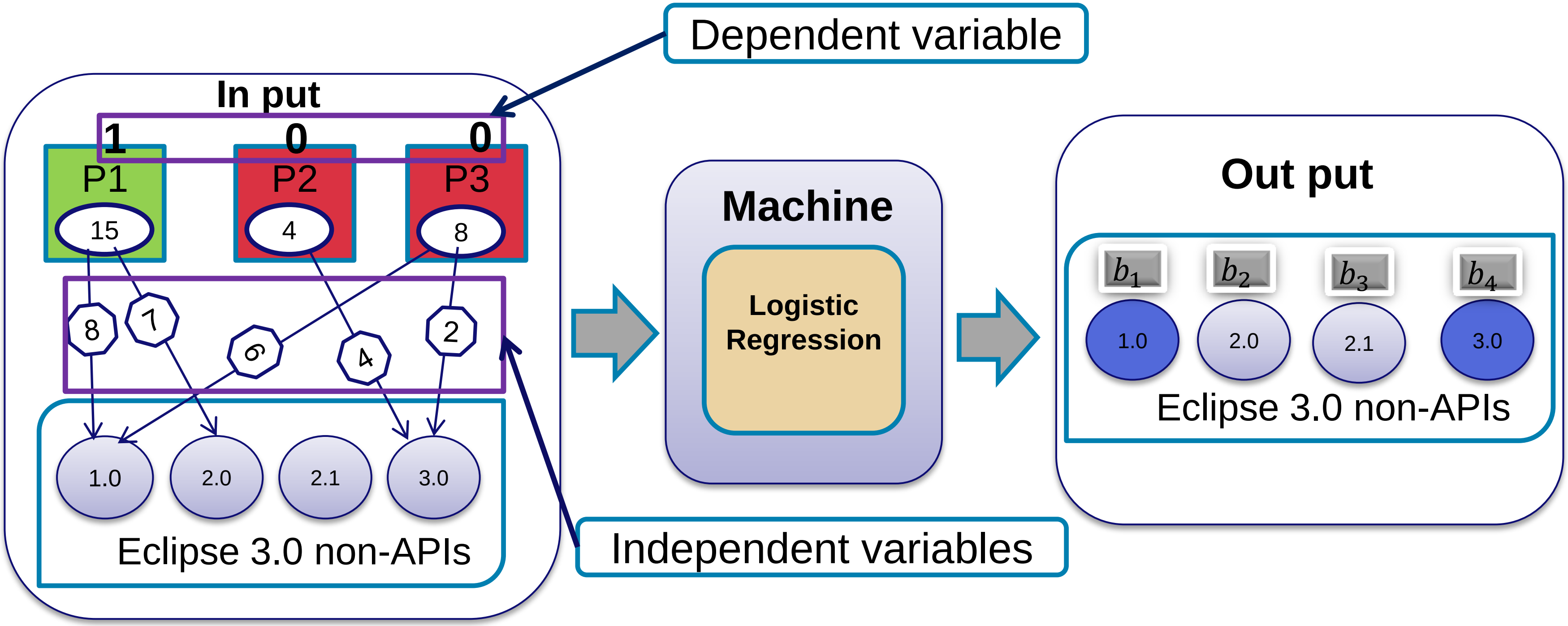
Model Training Example

ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0



Model Training Example

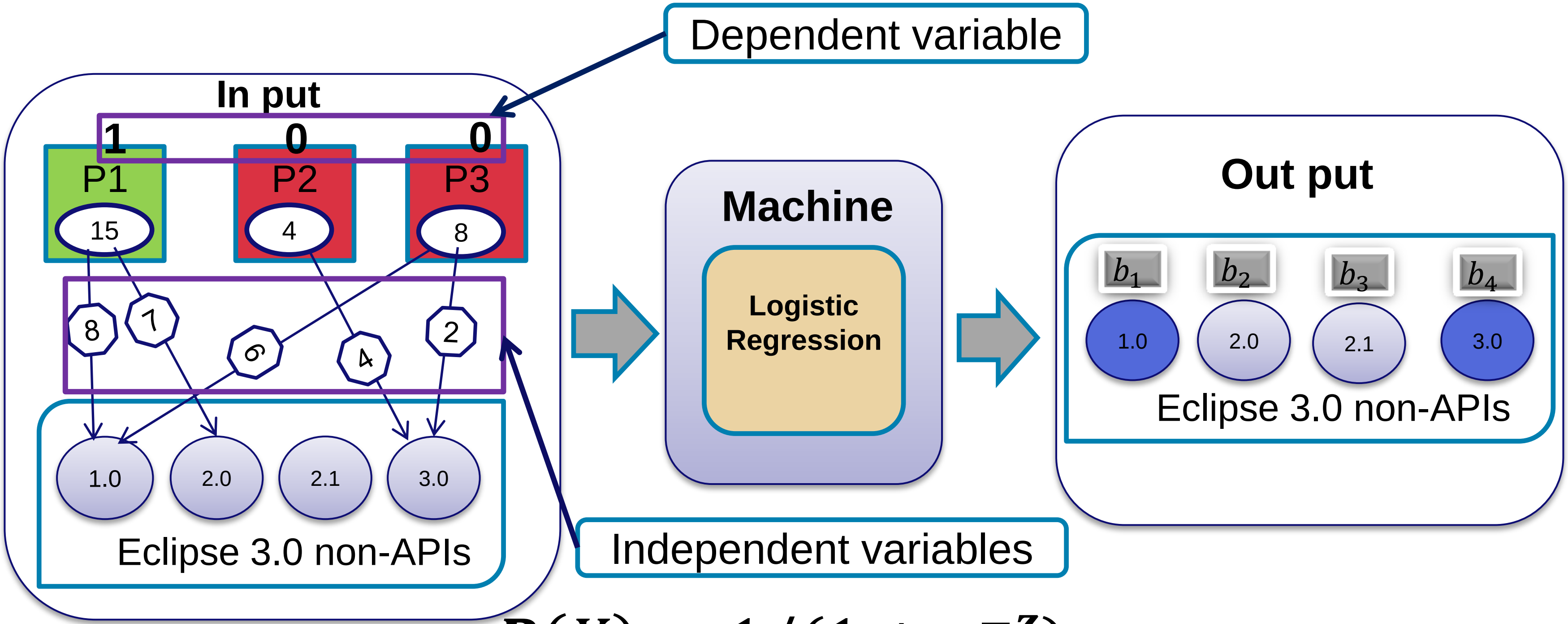
ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0



Technische Universiteit **Eindhoven** University of Technology

Model Training Example

ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0

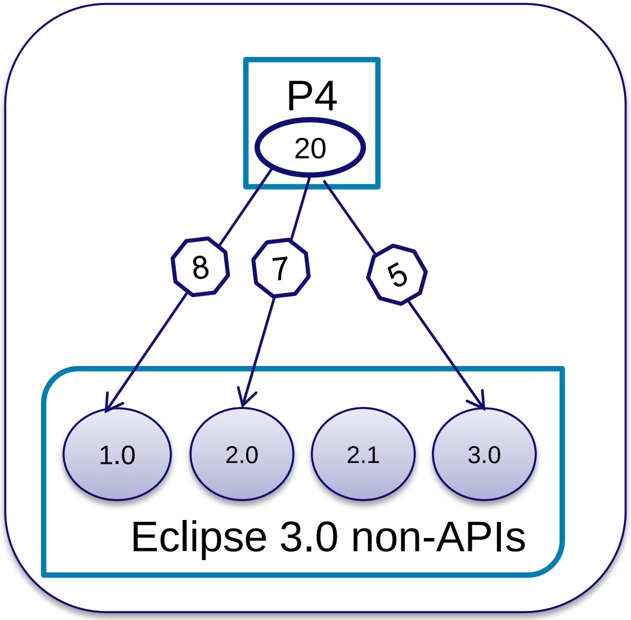


$$\boldsymbol{P}(X) = 1/(1 + e^{-\boldsymbol{z}})$$

$$\boldsymbol{z} = b_0 + b_1 * \boldsymbol{SDK}_{1.0} + b_4 * \boldsymbol{SDK}_{3.0}$$

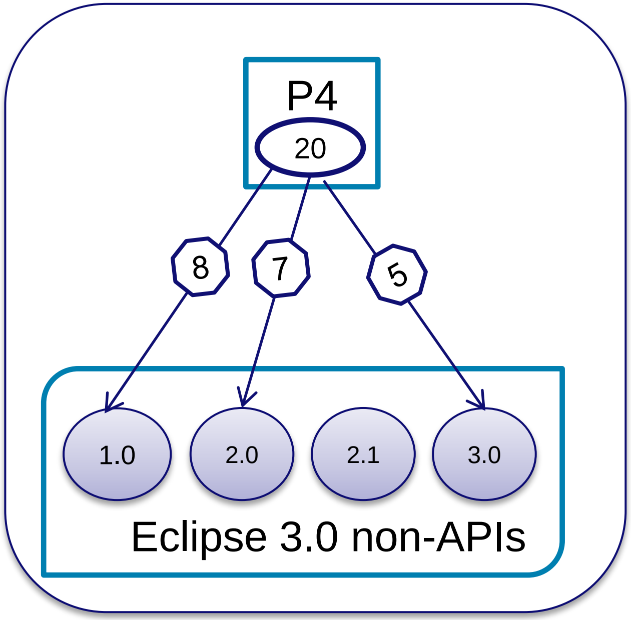
Model Training Example

ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0



Model Training Example

ETPs supported in Eclipse 3.0 – Prediction in Eclipse *X*, where *X* > 3.0



$\boldsymbol{P}(X) = 1/(1 + e^{-\boldsymbol{z}})$

$\boldsymbol{z} = b_0 + b_1 * 8 + b_4 * 5$

$\boldsymbol{P}(X)$<0.5 – incompatibility

$\boldsymbol{P}(X)$>=0.5 – compatibility

Results

- **In both model training and testing: High Precision, Accuracy, and Recall, where some were 80% and more**

	Model Testing Error Analysis								
	3.5			3.6			3.7		
	A	P	R	A	P	R	A	P	R
3.4	94	100	94	93	100	93	93	100	93
3.5				91	94	96	88	91	96

Conclusion and Future Work

- Mining interface usage from ETPs to detect or predict compatibility shows good results.
- Next, develop a domain specific tool to make predictions.
- Who can use the tool? users and developers of ETPs
-

Thank you for listening