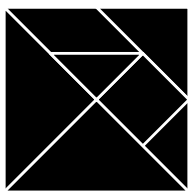
Usage Scenarios for Feature Model Synthesis.

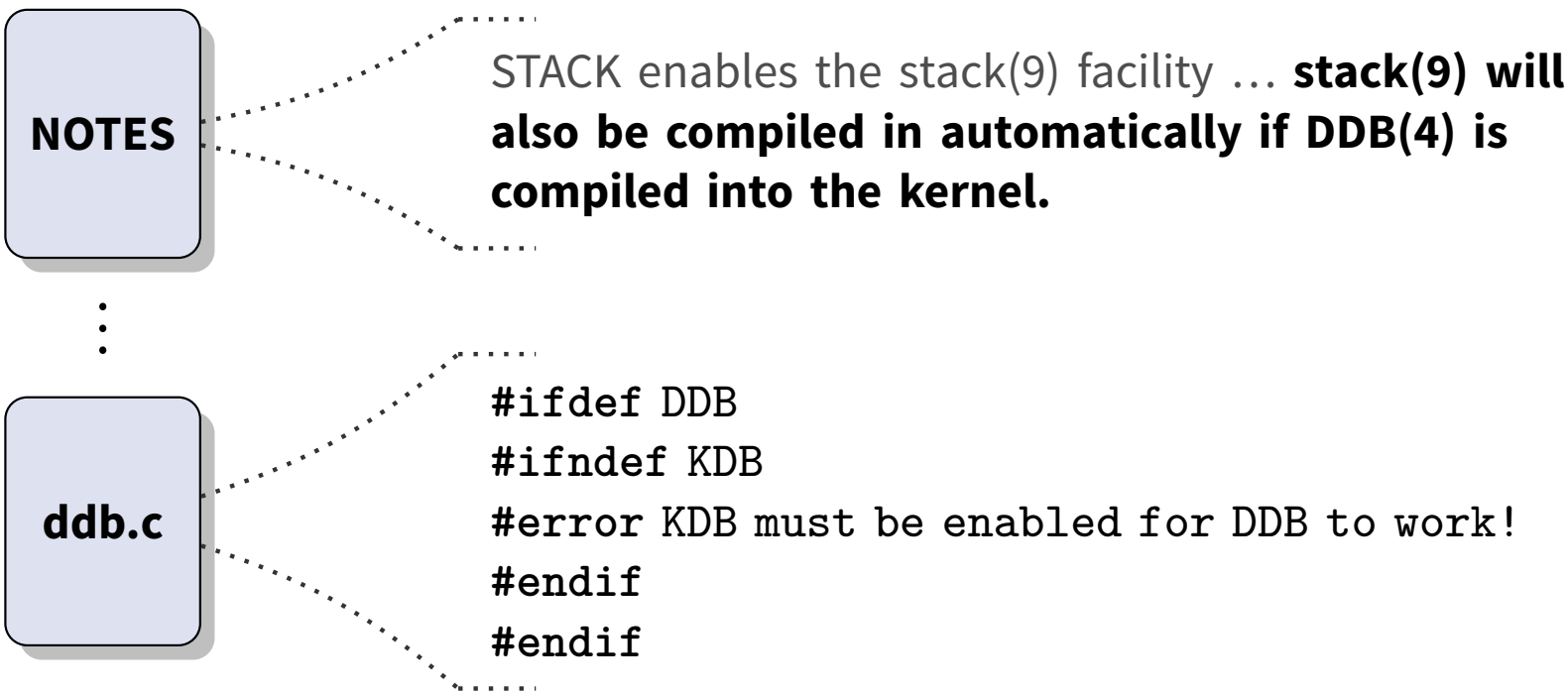


Steven She, Krzysztof Czarnecki, Andrzej Wasowski
University of Waterloo
IT University of Copenhagen

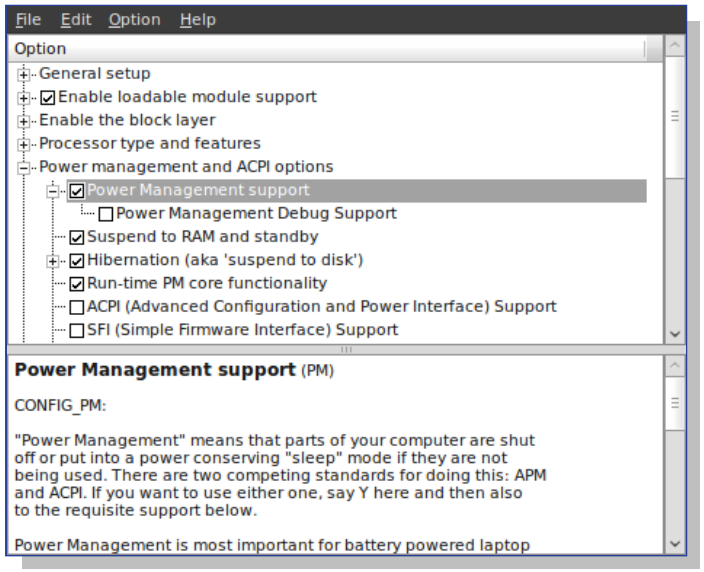
September 30, 2012.

Why Synthesize a Feature Model?

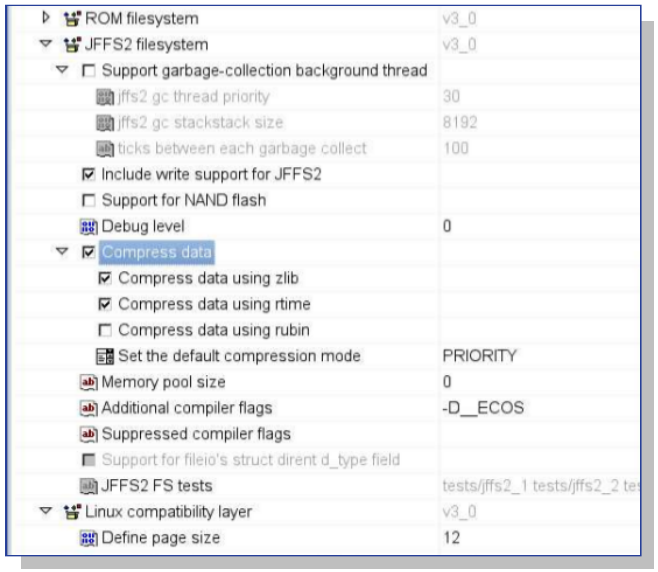
Variability is often scattered over multiple artifacts in large software projects.



Why Synthesize a Feature Model? (cont.)



Linux Configurator

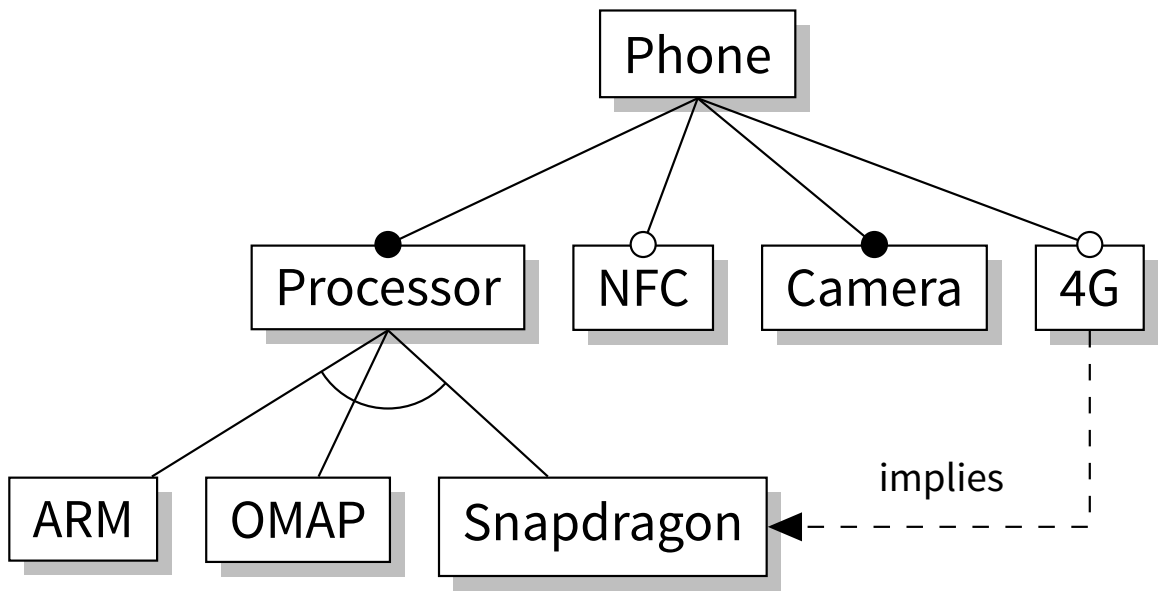


eCos Configurator

- A feature model provides an overview of variability in the software system.
- Automated tool support and product configuration.

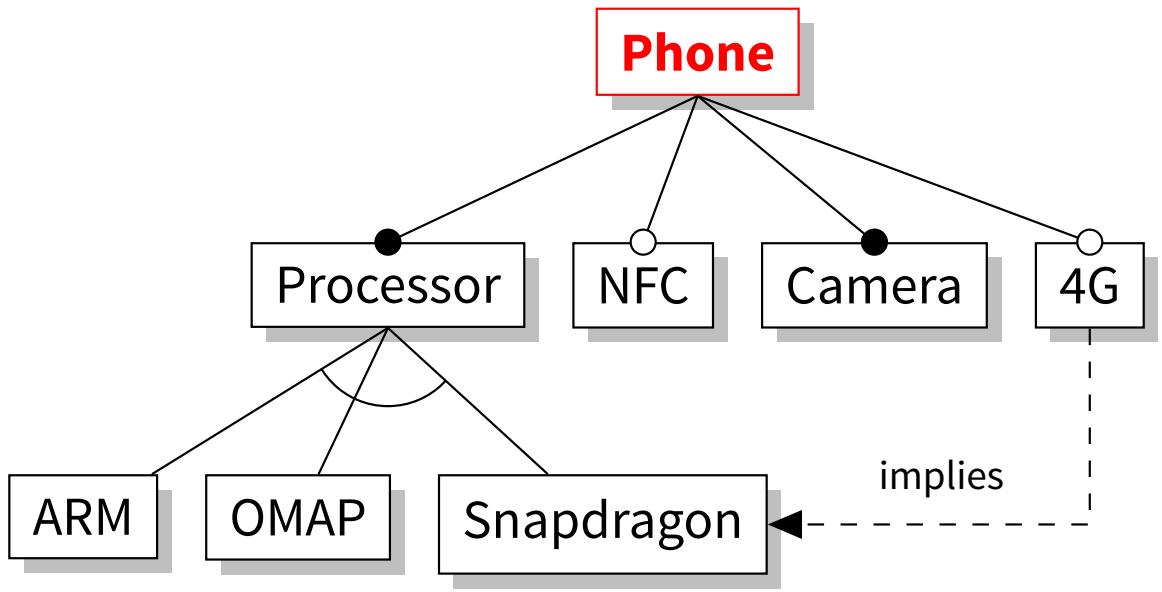
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.



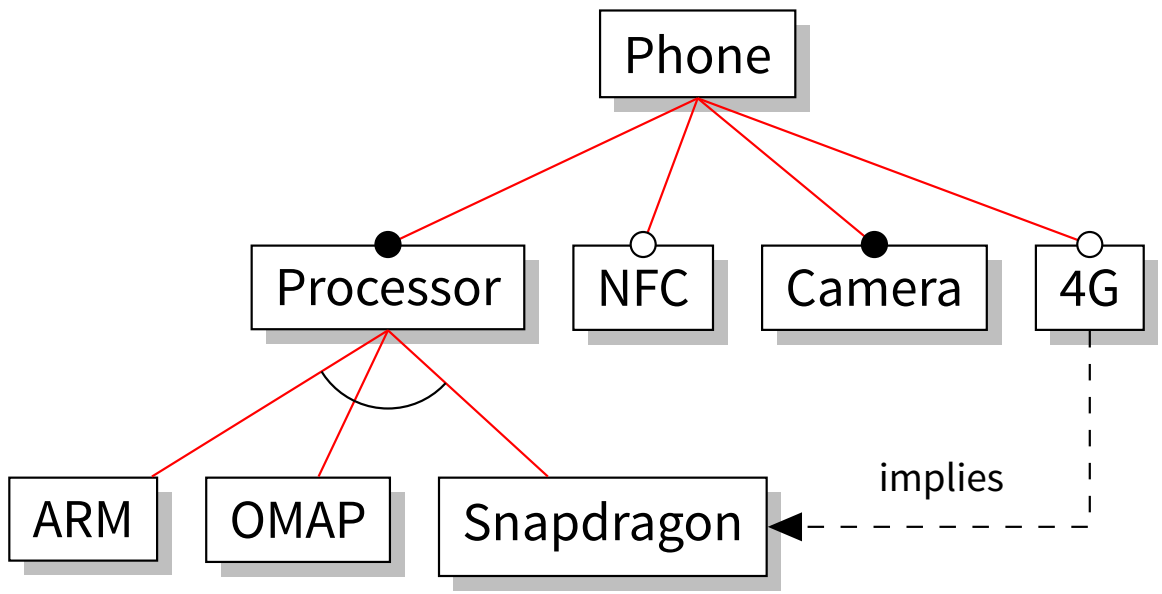
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.



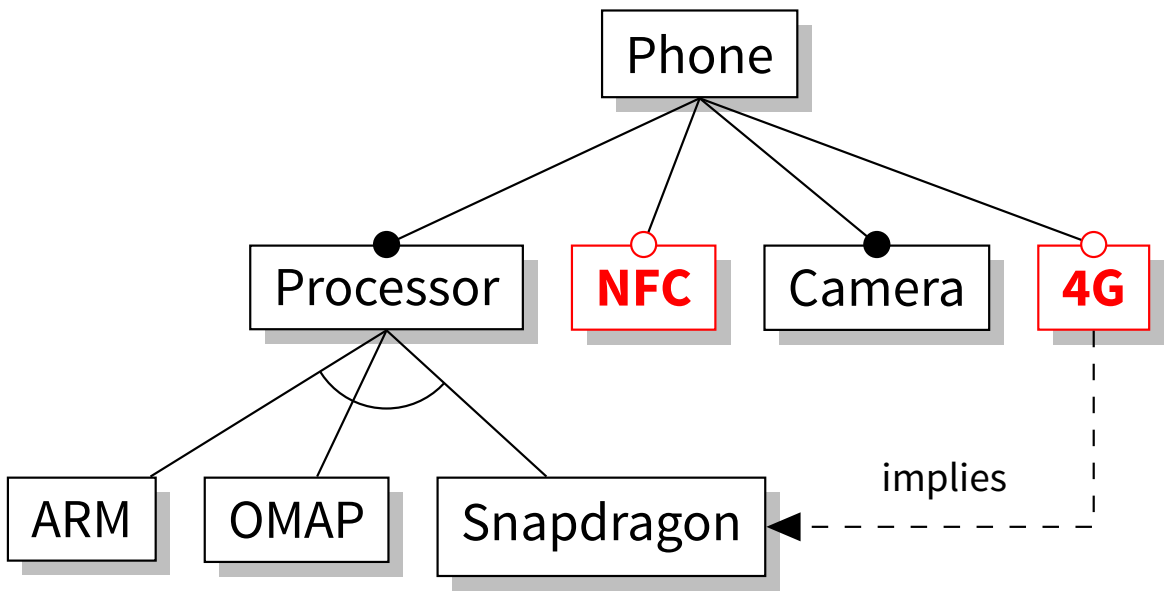
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.



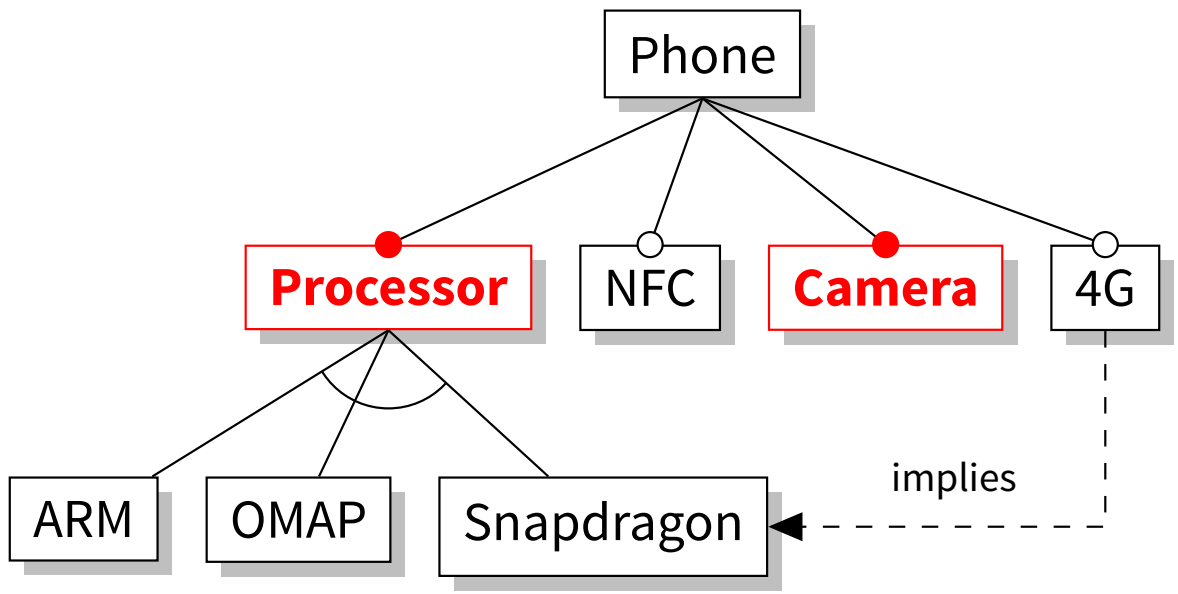
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.



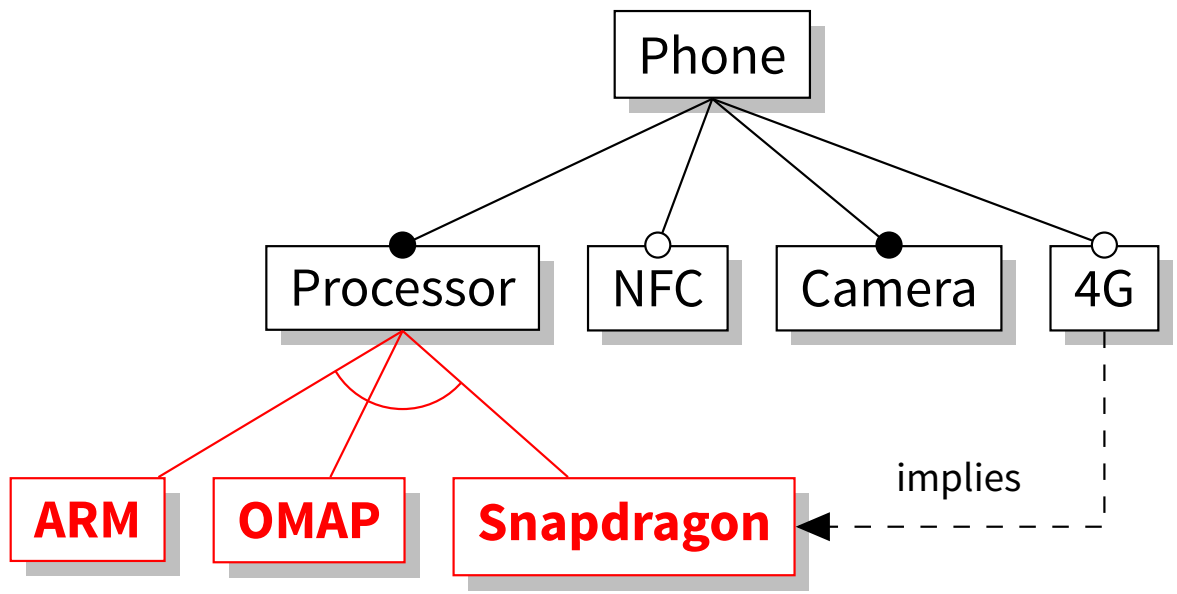
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.



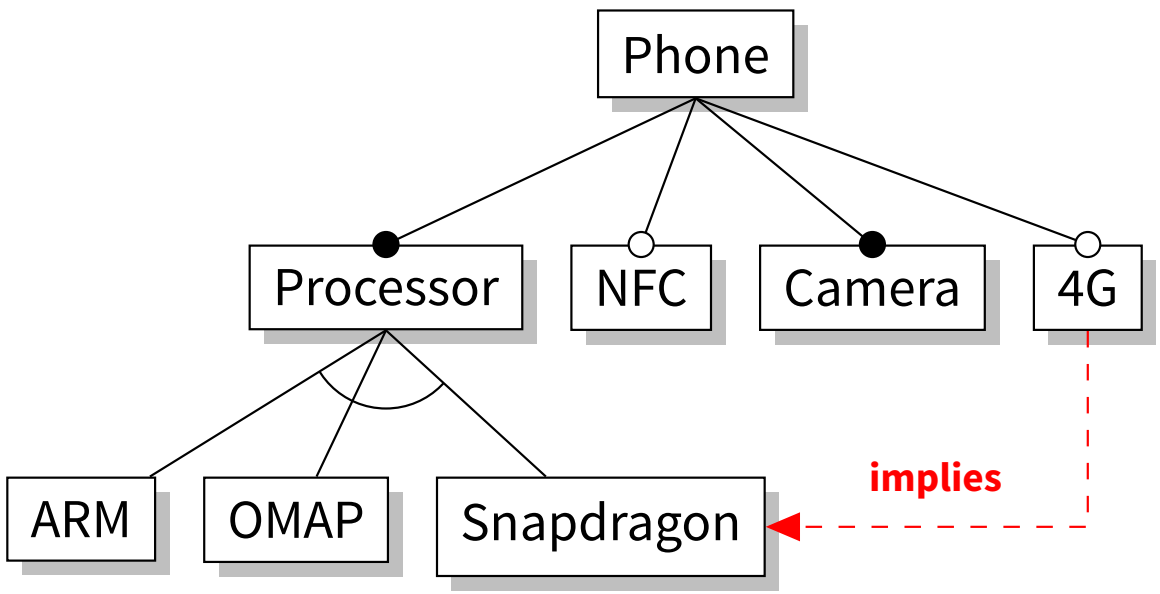
Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.

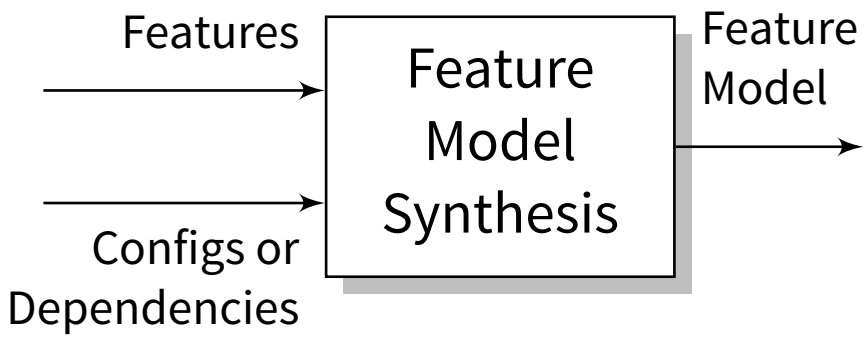


Feature Models.

Feature models describe the common and variable product characteristics of products in a product line.



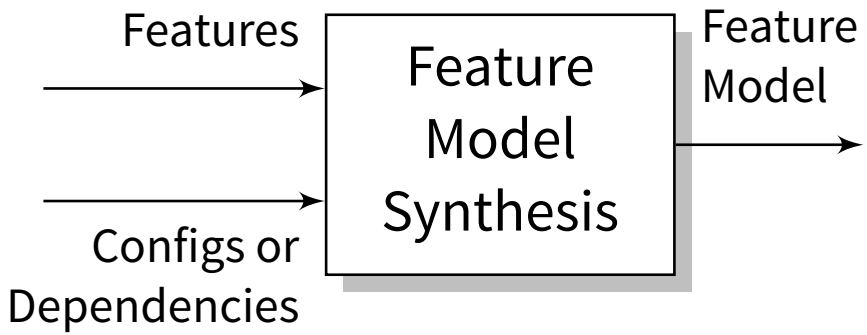
Feature Model Synthesis.



$\{\text{Phone}, \text{Processor}, \text{ARM}, \text{Snapdragon}, \text{NFC}\}$

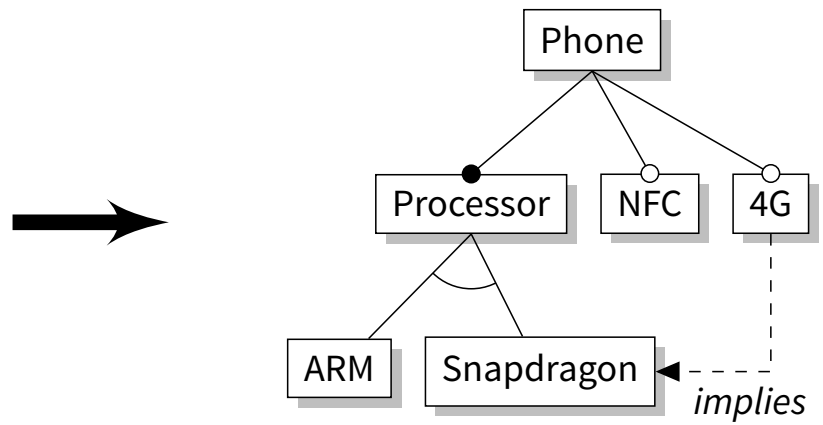
$$
\begin{aligned}
&(4G \wedge NFC \rightarrow Phone)\\
\wedge\ &(Processor \leftrightarrow Phone)\\
\wedge\ &(ARM \wedge Snapdragon \rightarrow Processor)\\
\wedge\ &(4G \rightarrow Snapdragon)\\
\wedge\ &(ARM \rightarrow \neg Snapdragon)
\end{aligned}
$$

Feature Model Synthesis.

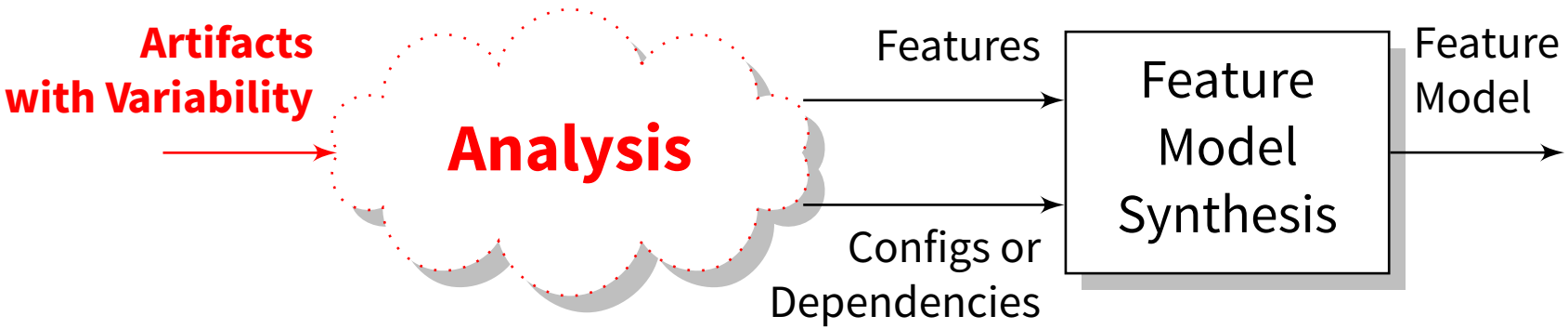


$\{\text{Phone}, \text{Processor}, \text{ARM}, \text{Snapdragon}, \text{NFC}\}$

$$
\begin{aligned}
& (4G \wedge NFC \rightarrow Phone) \\
\wedge\ & (Processor \leftrightarrow Phone) \\
\wedge\ & (ARM \wedge Snapdragon \rightarrow Processor) \\
\wedge\ & (4G \rightarrow Snapdragon) \\
\wedge\ & (ARM \rightarrow \neg Snapdragon)
\end{aligned}
$$

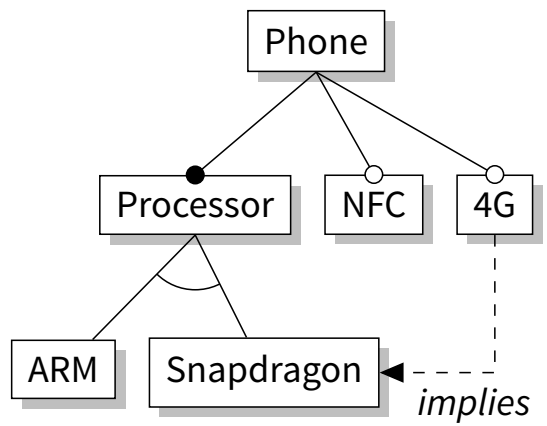


Feature Model Synthesis.

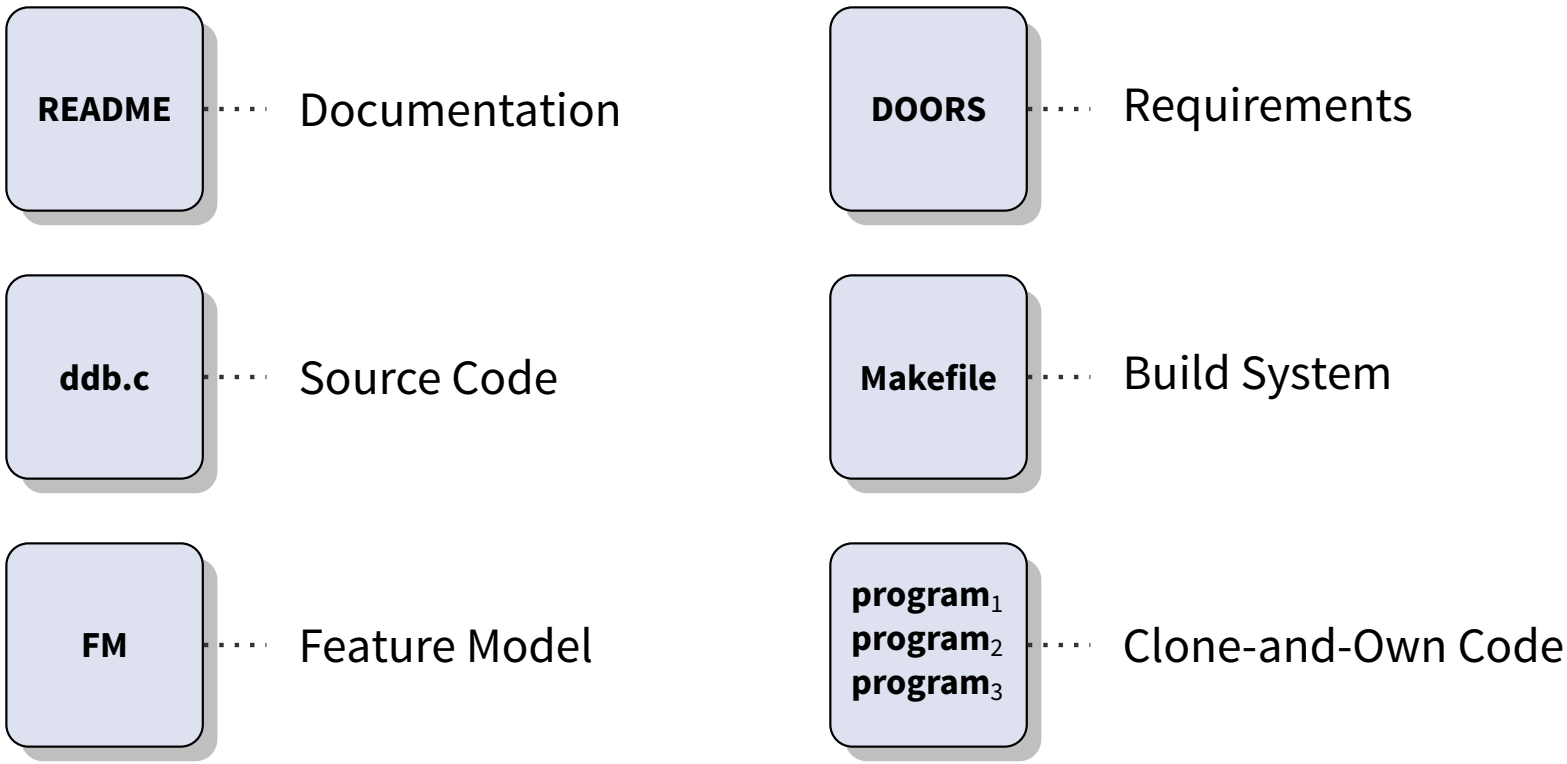


$\{\text{Phone}, \text{Processor}, \text{ARM}, \text{Snapdragon}, \text{NFC}\}$

$$
\begin{aligned}
& (4G \wedge NFC \rightarrow Phone) \\
\wedge\ & (Processor \leftrightarrow Phone) \\
\wedge\ & (ARM \wedge Snapdragon \rightarrow Processor) \\
\wedge\ & (4G \rightarrow Snapdragon) \\
\wedge\ & (ARM \rightarrow \neg Snapdragon)
\end{aligned}
$$

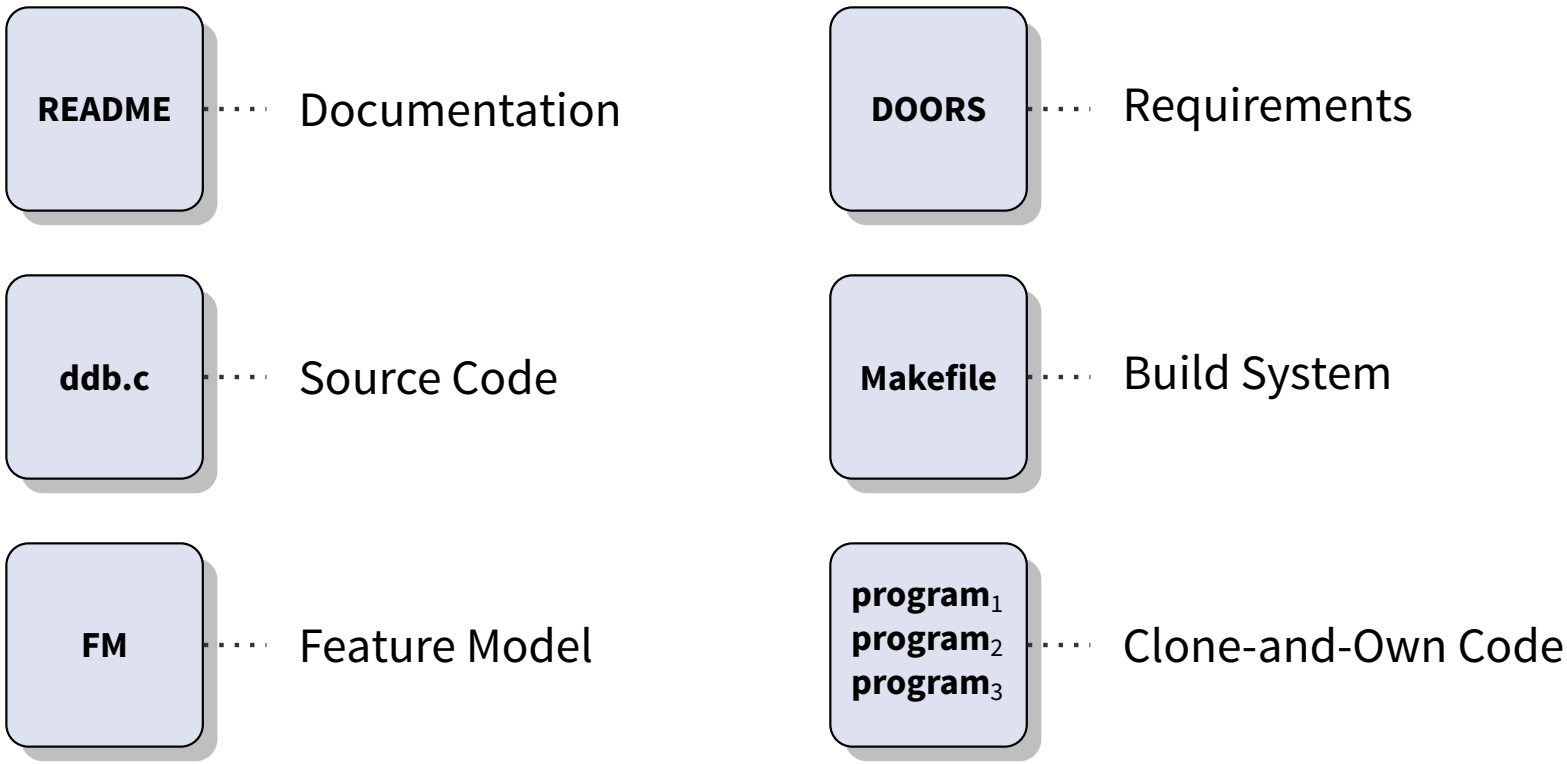


Artifacts with Variability.



…and more.

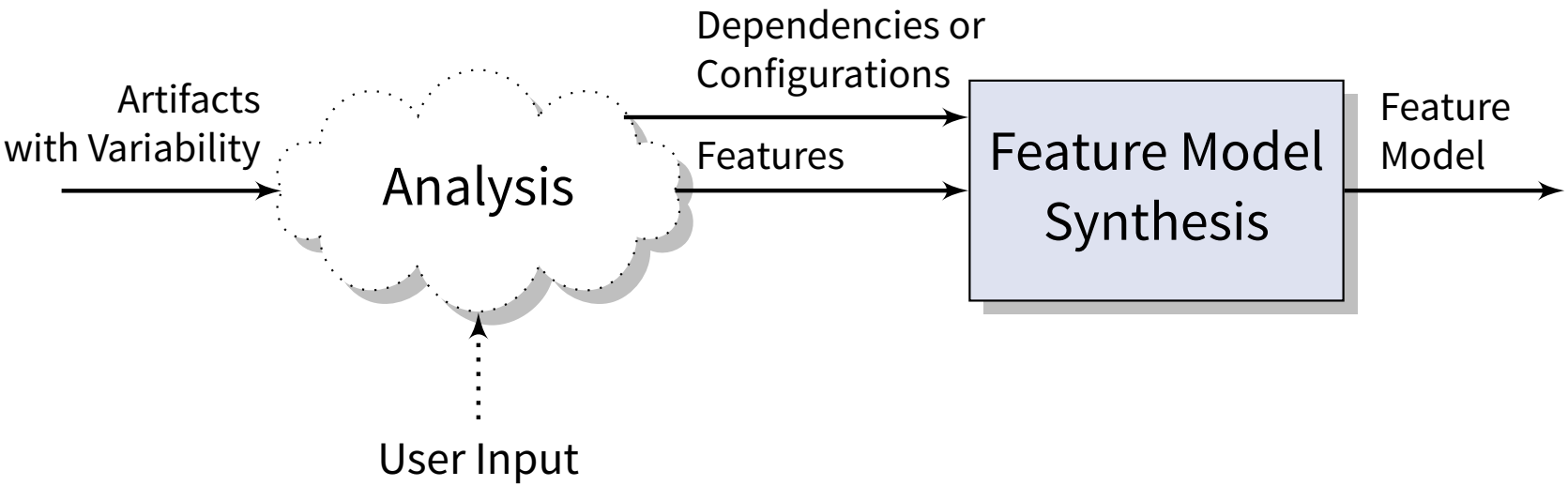
Artifacts with Variability.



…and more.

Many different kinds of input artifacts,
each requiring specialized analysis.

Workflow for Feature Model Synthesis.

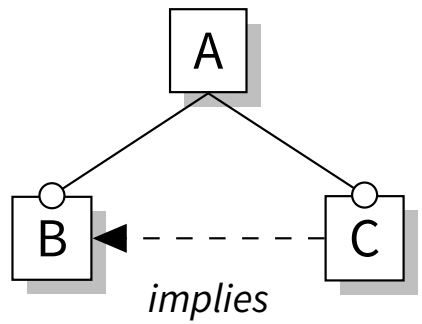


- **Analysis** recovers the abstract input needed for feature model synthesis from the input artifacts.
- **Feature model synthesis** builds a FM given the abstract input.

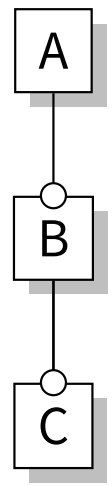
Challenges of Feature Model Synthesis.

Which feature model describes features $\{A, B, C\}$, and the following constraints:

$$B \rightarrow A$$
$$C \rightarrow A$$
$$C \rightarrow B$$



(a)



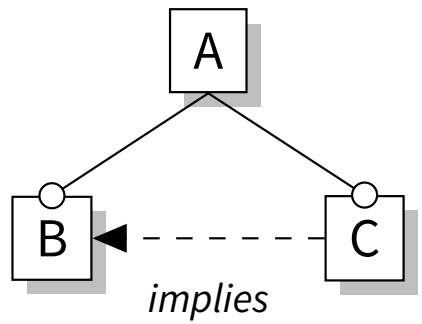
(b)

Both! Different feature models can describe the same set of configurations (or dependencies).

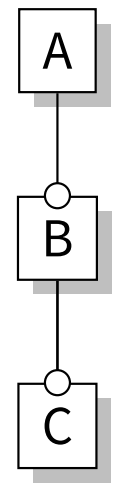
Challenges of Feature Model Synthesis.

Which feature model describes features $\{A, B, C\}$, and the following constraints:

$$B \rightarrow A$$
$$C \rightarrow A$$
$$C \rightarrow B$$



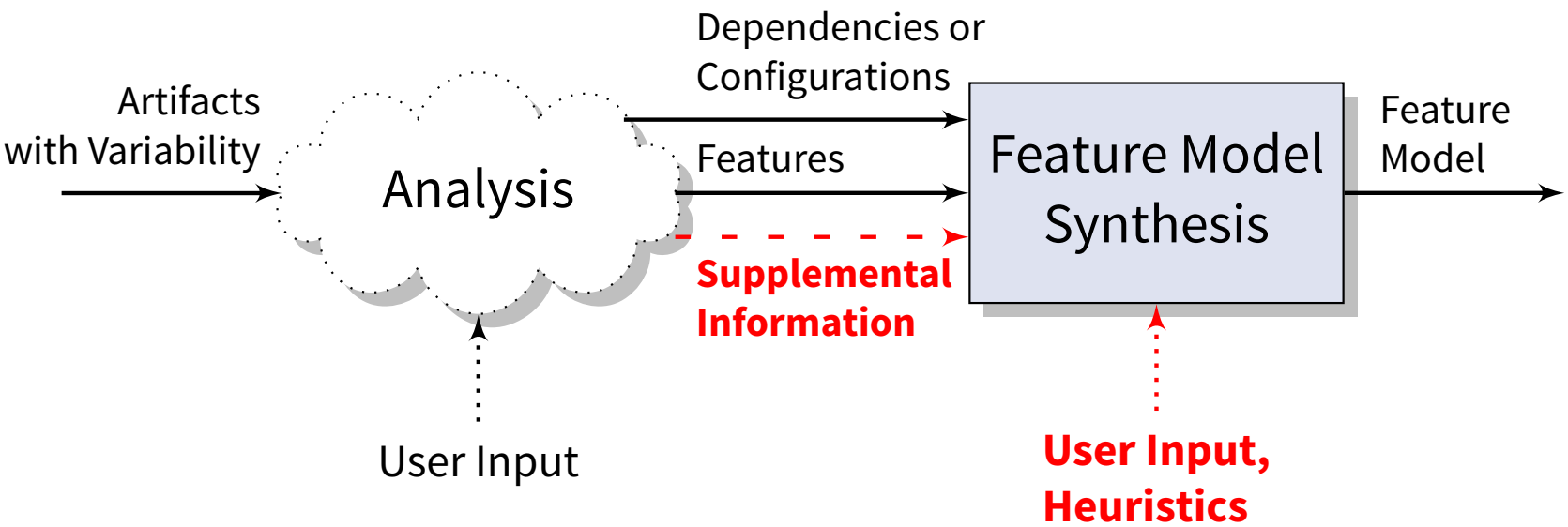
(a)



(b)

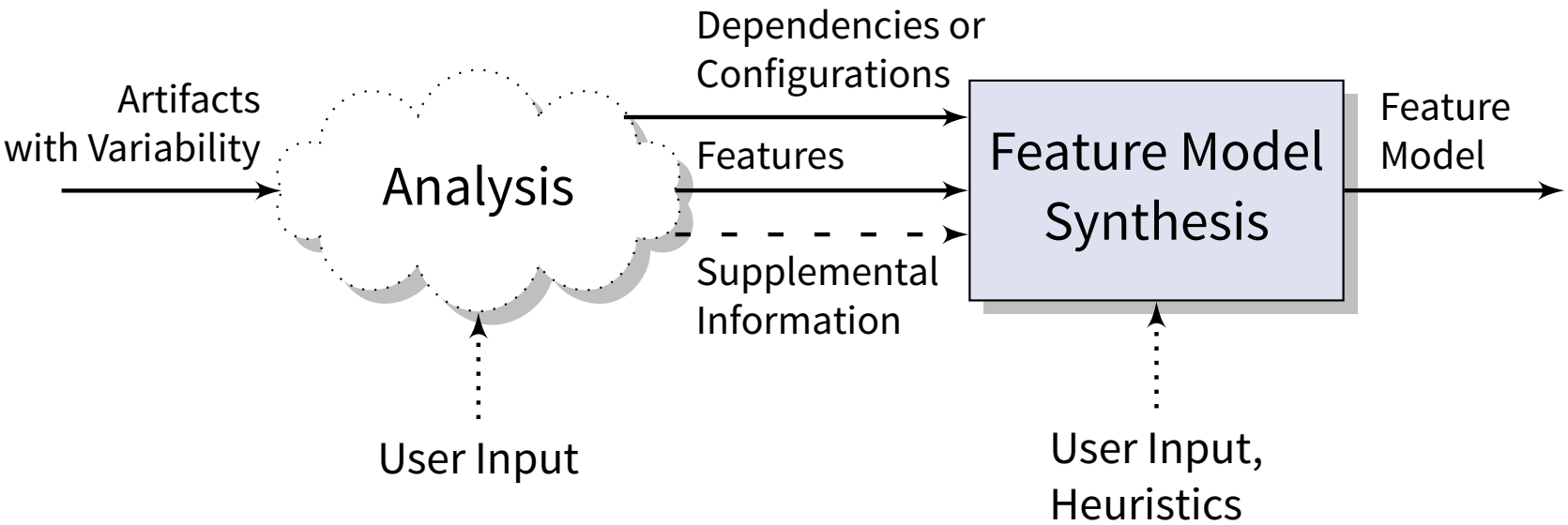
Both! Different feature models can describe the same set of configurations (or dependencies).

Workflow for Feature Model Synthesis.



- Supplemental information, user input or heuristics can be used to select a distinct feature model.

Workflow for Feature Model Synthesis.



Outline.

Outline.

Scenario Criteria.

1. Input Artifacts
2. Precision of Configuration Analysis
3. Required Synthesis Precision
4. Size

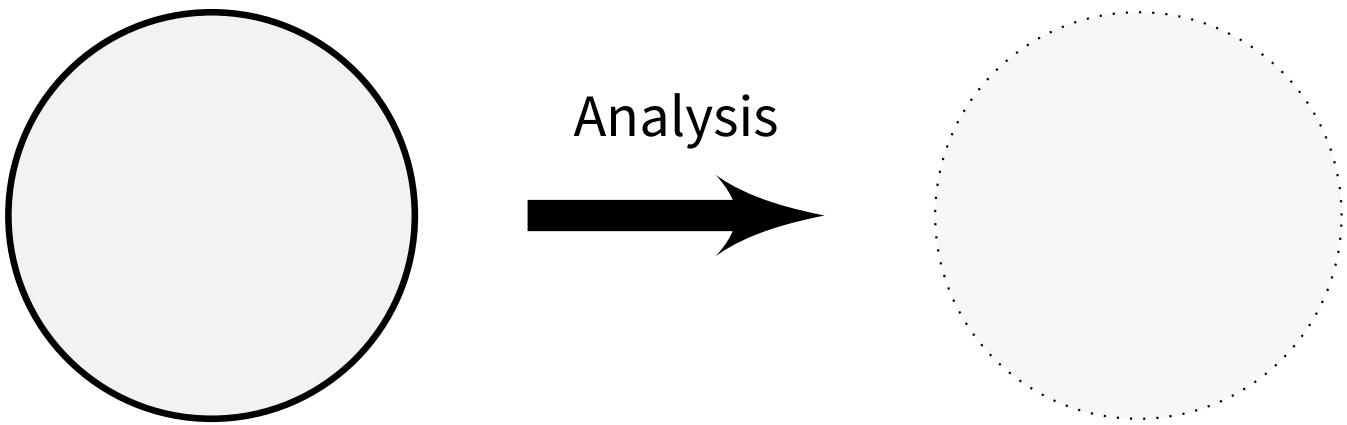
Input Artifacts.

	Variability Abstraction	Abstraction-Realization Interface	Variability Realization
Variable Artifacts	Feature model	VPs and feature-to-VP mapping	Configurable platform *requirements, models, code, etc.*
Instances	Feature configs.	VP configs.	Variants *requirements, models, code, etc.*

Precision of Configuration Analysis.

Configurations represented by the input artifacts

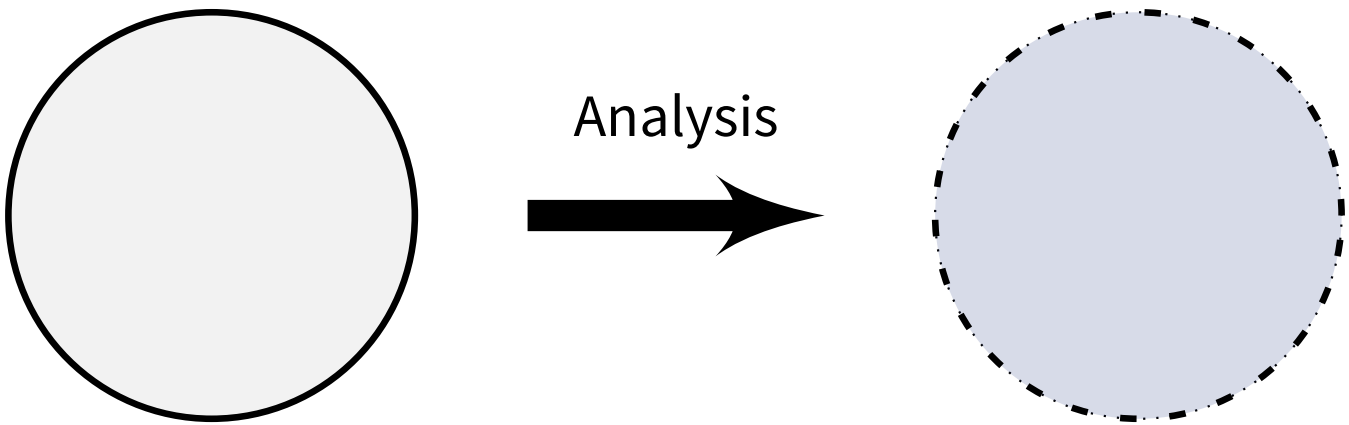
Configurations recovered through analysis



Precision of Configuration Analysis.

Configurations represented by the input artifacts

Configurations recovered through analysis

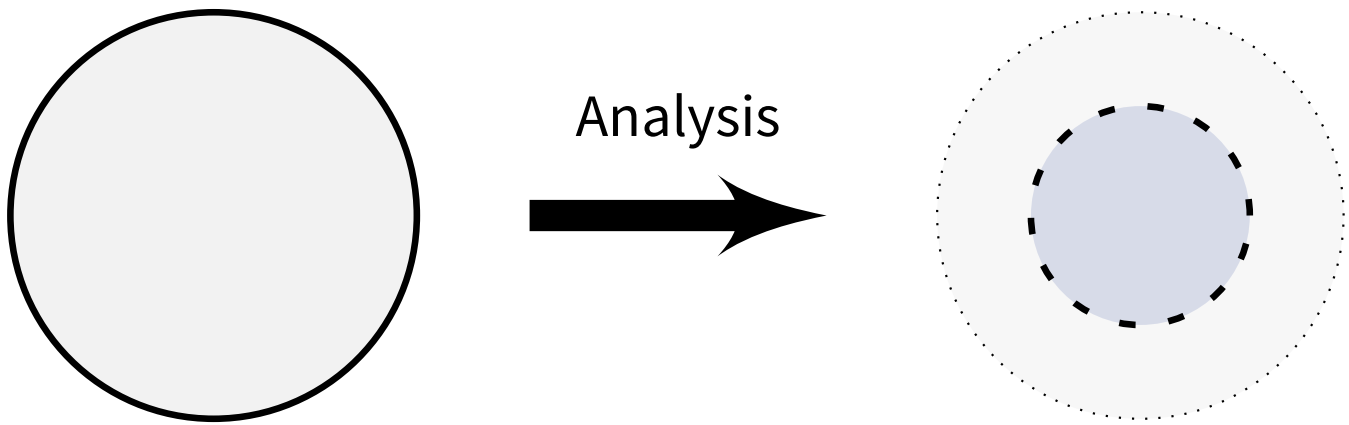


Sound and complete (exact) recovery.

Precision of Configuration Analysis.

Configurations represented by the input artifacts

Configurations recovered through analysis

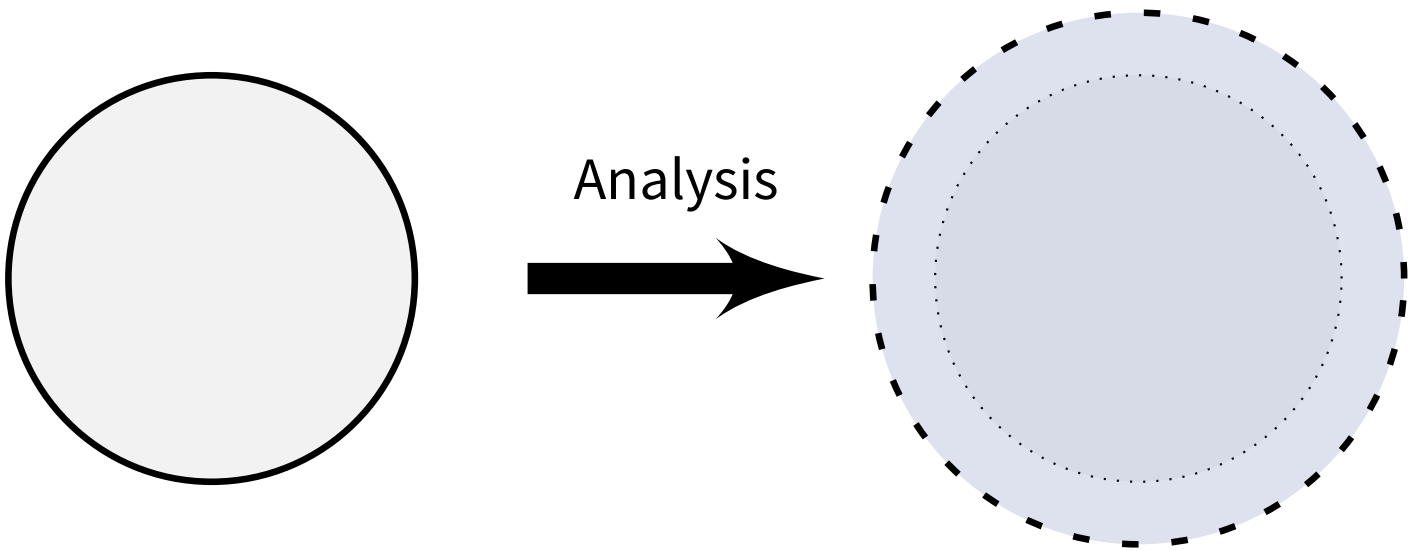


Sound recovery (under approximation).

Precision of Configuration Analysis.

Configurations represented by the input artifacts

Configurations recovered through analysis

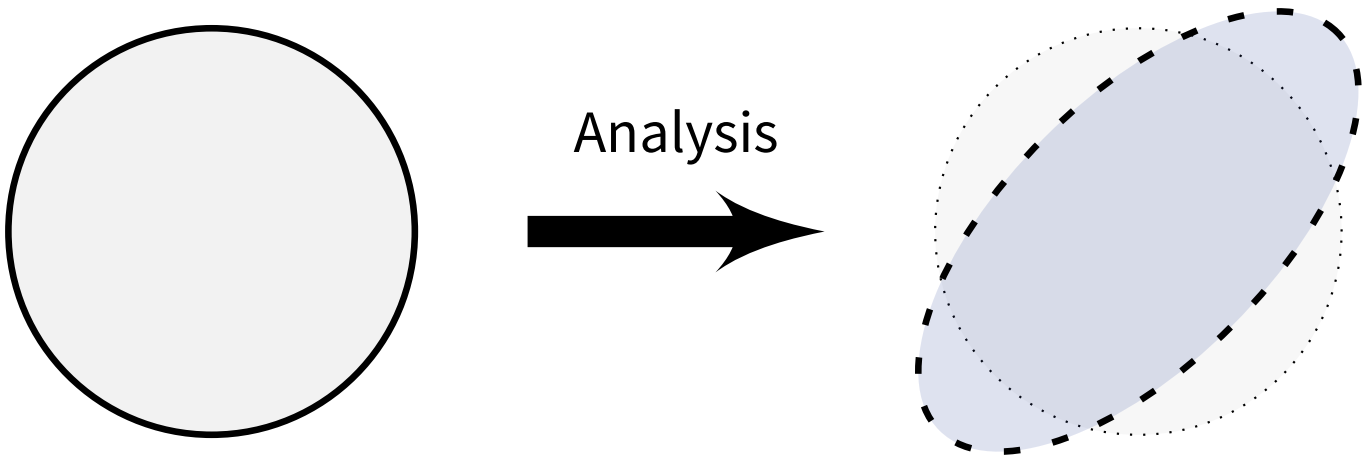


Complete recovery (over approximation).

Precision of Configuration Analysis.

Configurations represented by the input artifacts

Configurations recovered through analysis



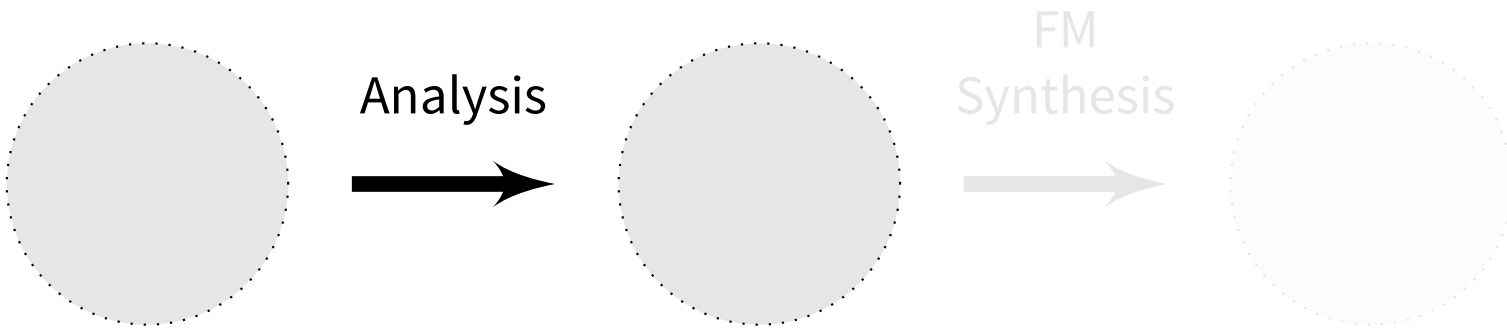
Arbitrary recovery.

Required Synthesis Precision.

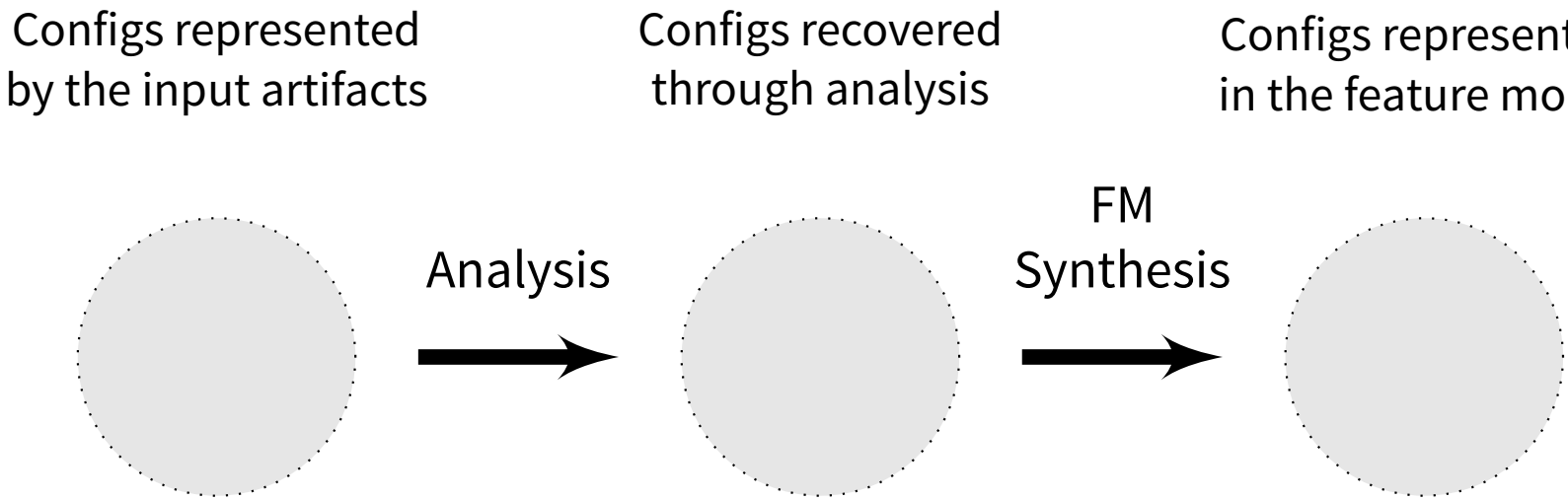
Configs represented by the input artifacts

Configs recovered through analysis

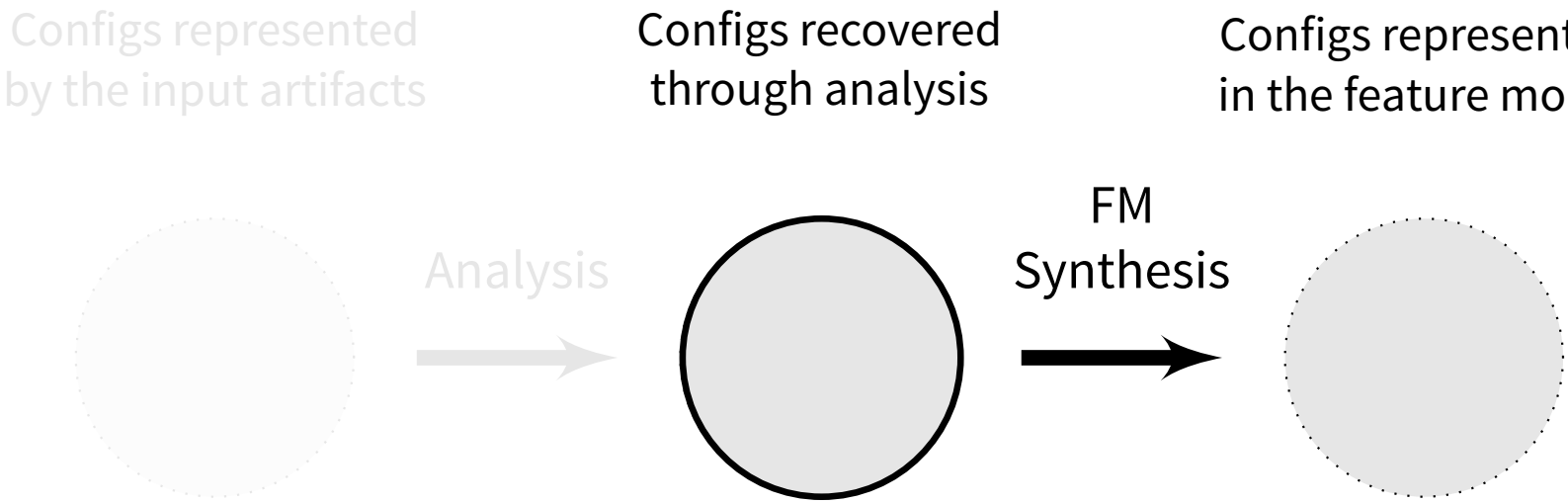
Configs represented in the feature model



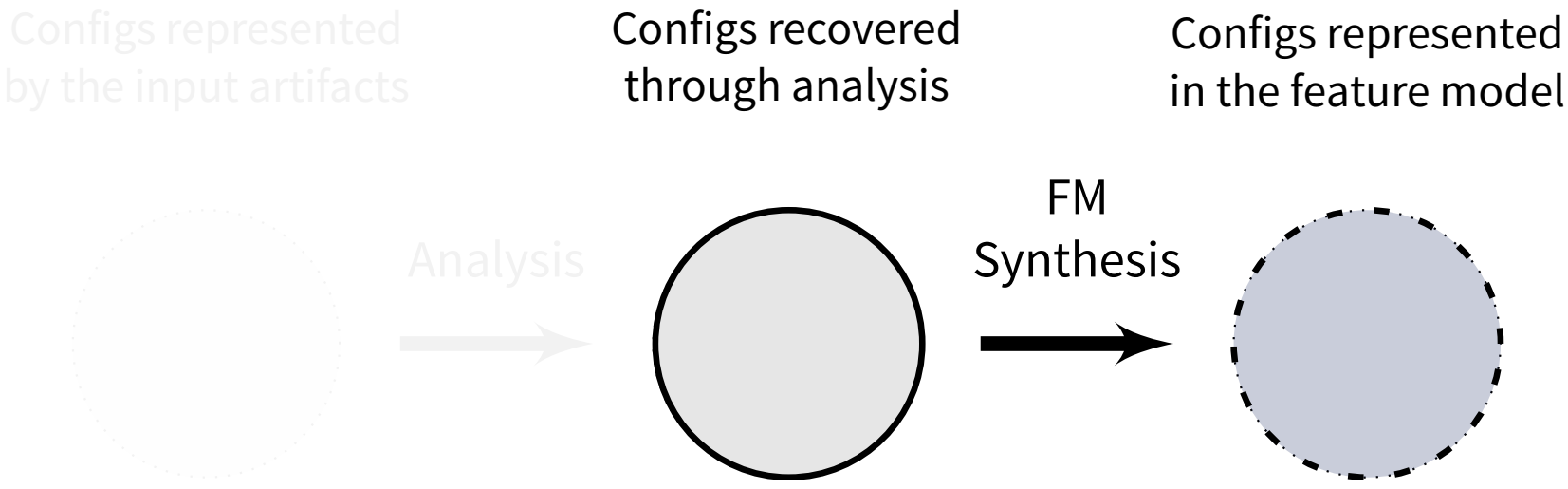
Required Synthesis Precision.



Required Synthesis Precision.

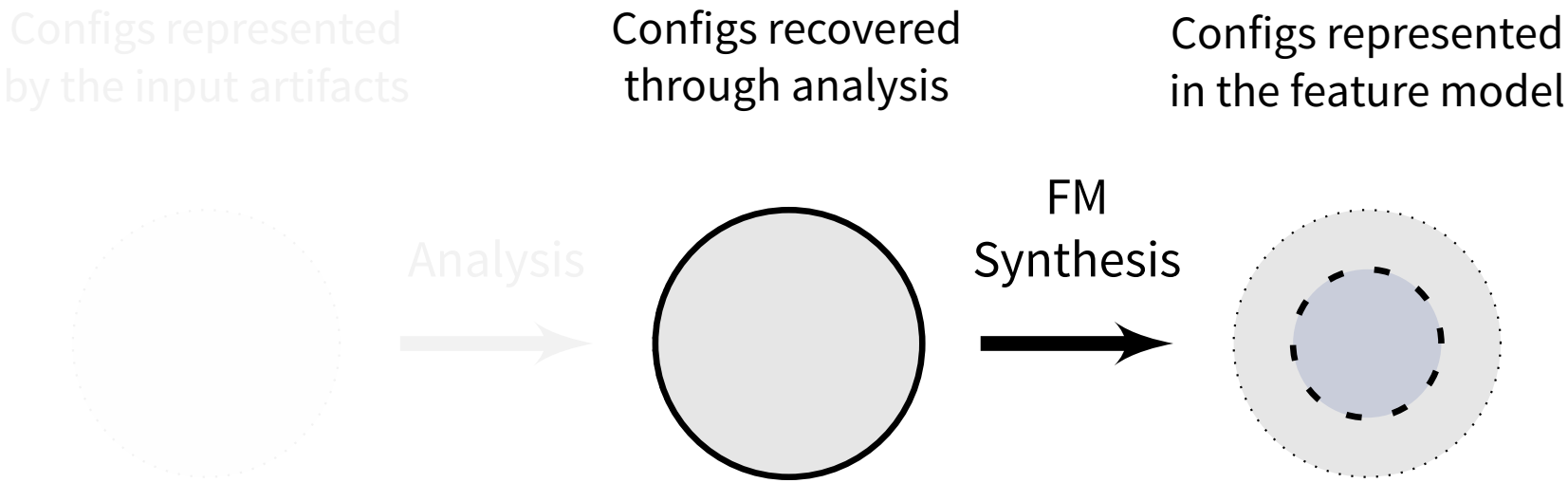


Required Synthesis Precision.



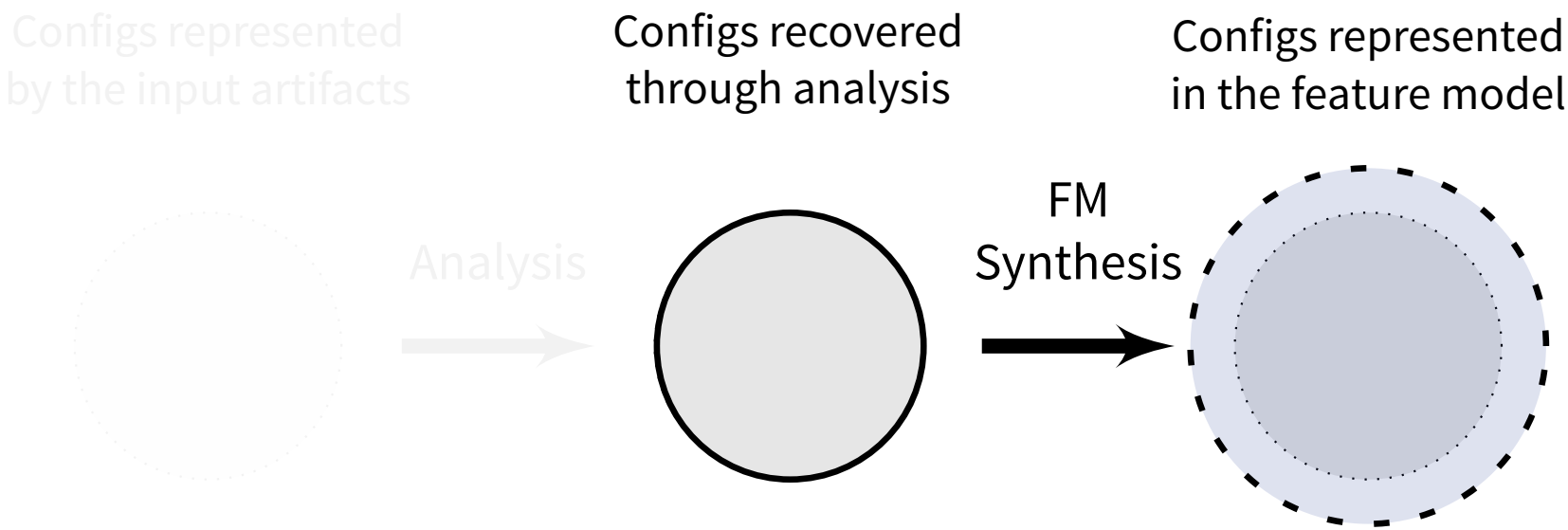
Sound and complete (exact) recovery.

Required Synthesis Precision.



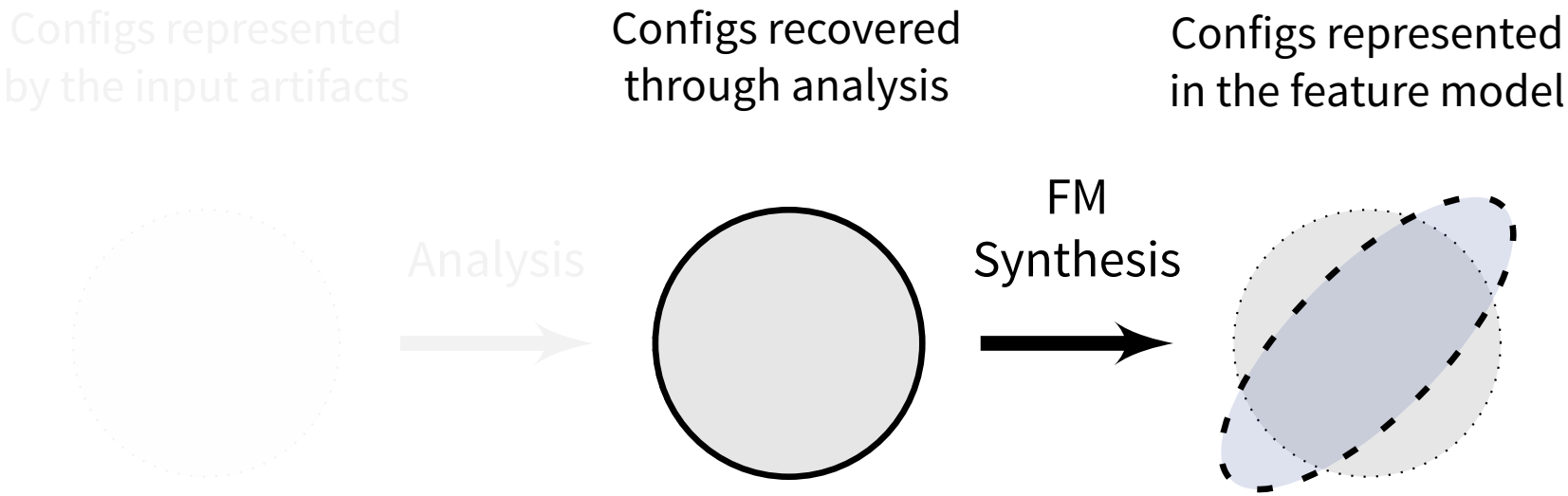
Sound recovery (under approximation).

Required Synthesis Precision.



Complete recovery (over approximation).

Required Synthesis Precision.



Arbitrary recovery.

Size.

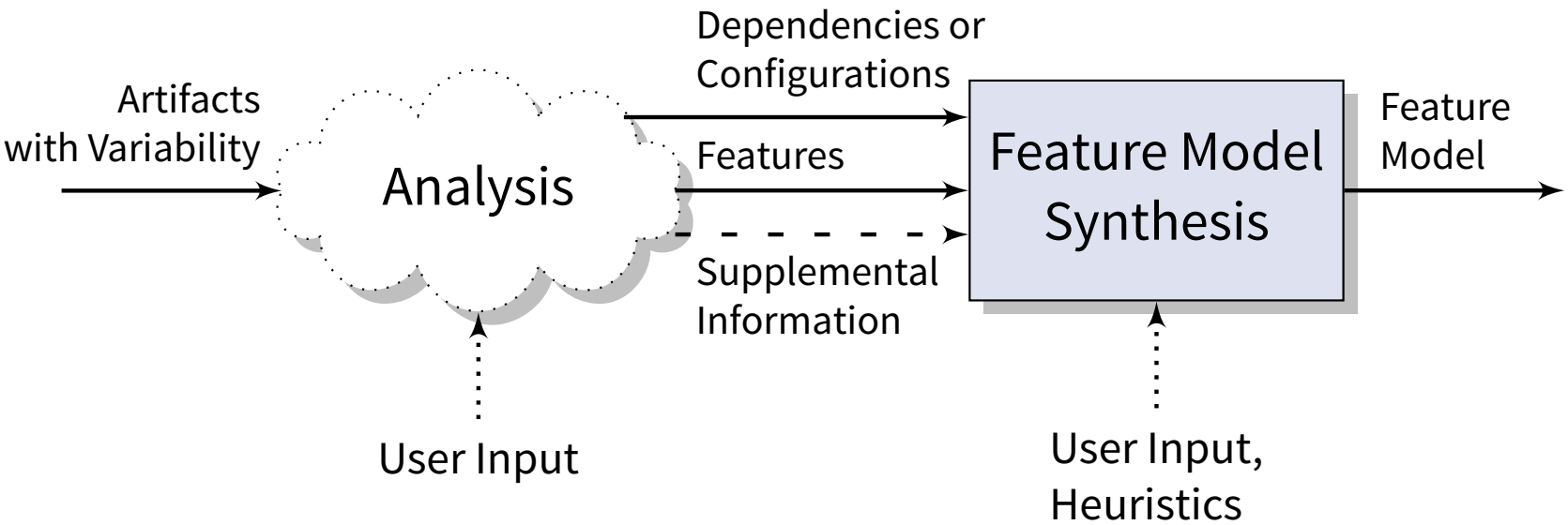
- Classified size of a scenario by estimating the number of features required for synthesis.
- Based on SPLOT's model repository[1] and a collection of FMs gathered from the system's domain.

Small	Several hundred features.
Medium	Thousand features.
Large	Several thousand features.

[1] `http://www.splot-research.org`

Outline.

Workflow for Feature Model Synthesis.

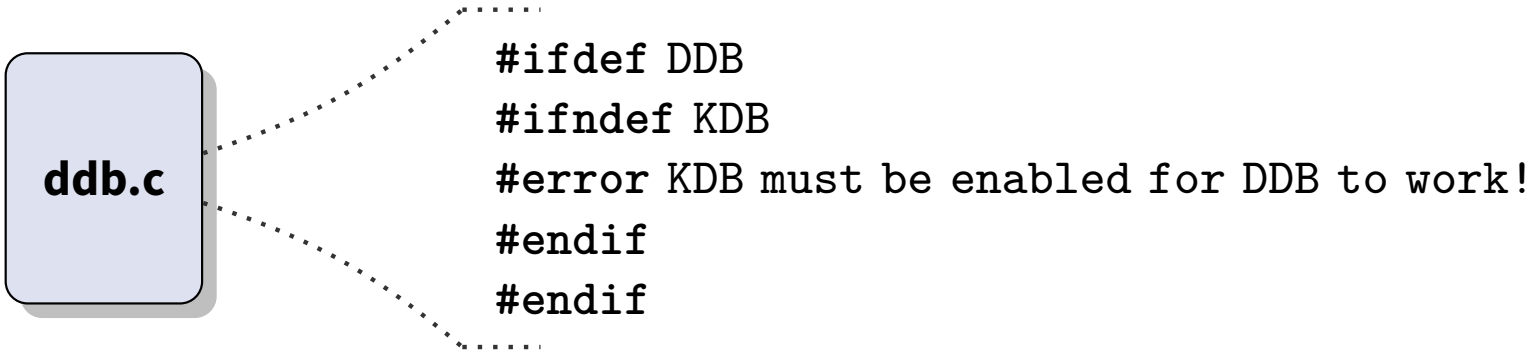


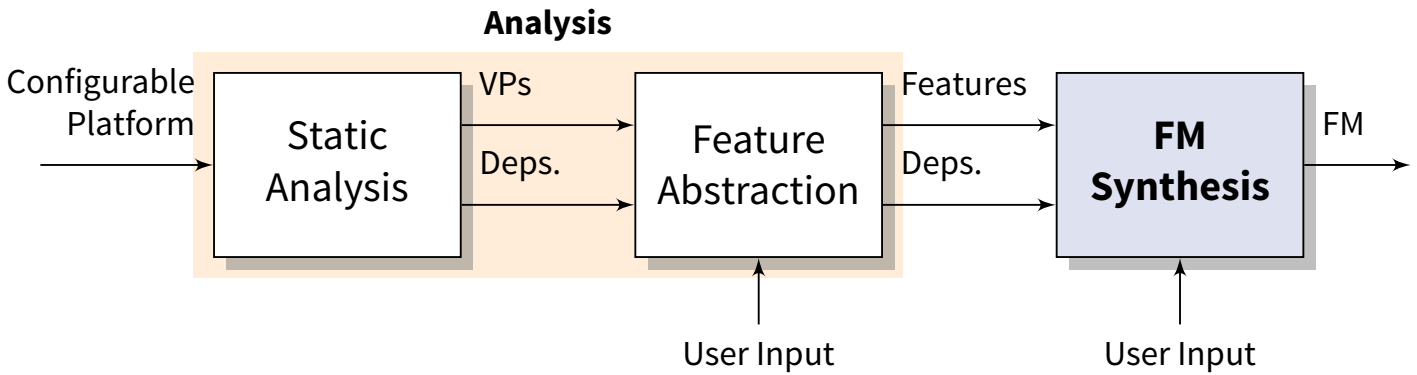
Scenario Outline.

Scenario 1: From a Configurable Platform

	Variability Abstraction	Abstraction-Realization Interface	Variability Realization
Variable Artifacts	Feature model	VPs and feature-to-VP mapping	**Configurable platform** ***requirements, models, code, etc.***
Instances	Feature configs.	VP configs.	Variants *requirements, models, code, etc.*

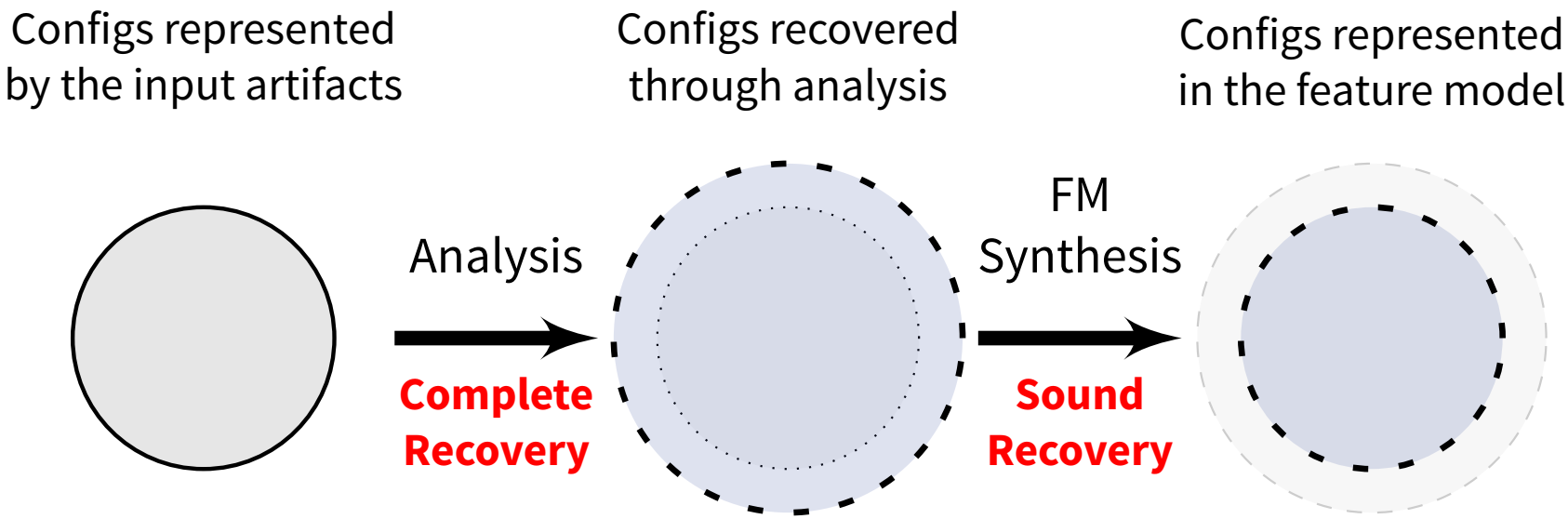
Configurable Platform (Code).



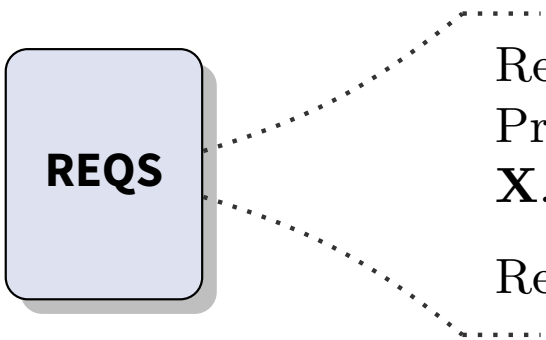


- Scenario for synthesizing a FM for FreeBSD (She et al. 2011)

Configurable Platform (Code): Precision.



Configurable Platform (Requirements).



Req. 1. Product A must have **Feature X**, while Product B may optionally implement **Feature X**. Every product must have **Feature Y**.

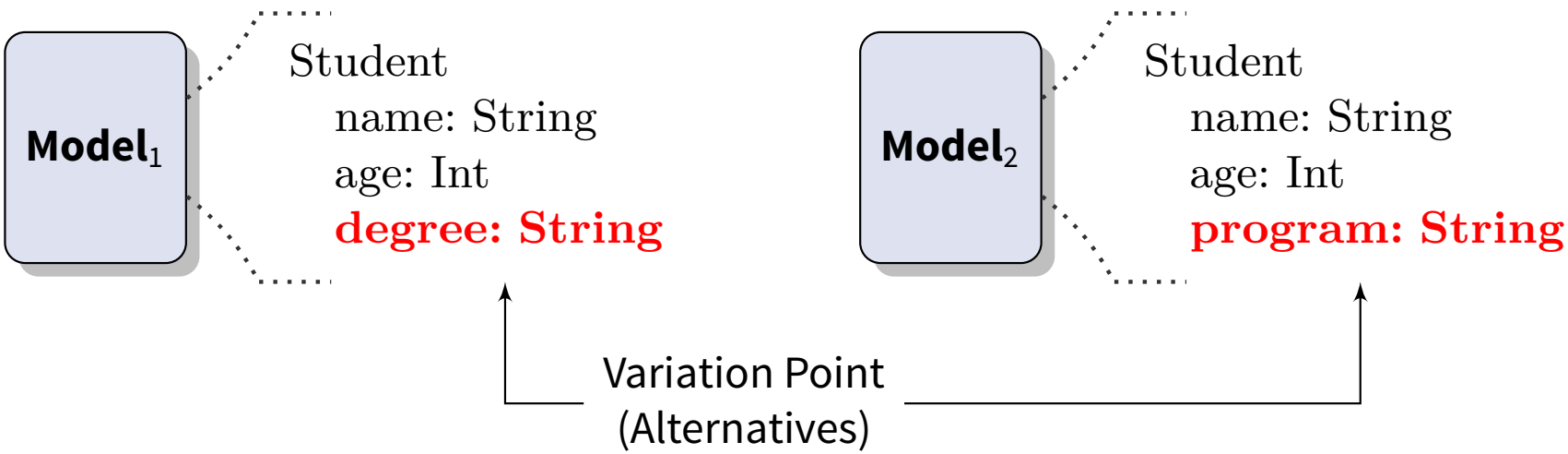
Req. 2. ...

Scenario Outline.

Scenario 2: From a Set of Variants

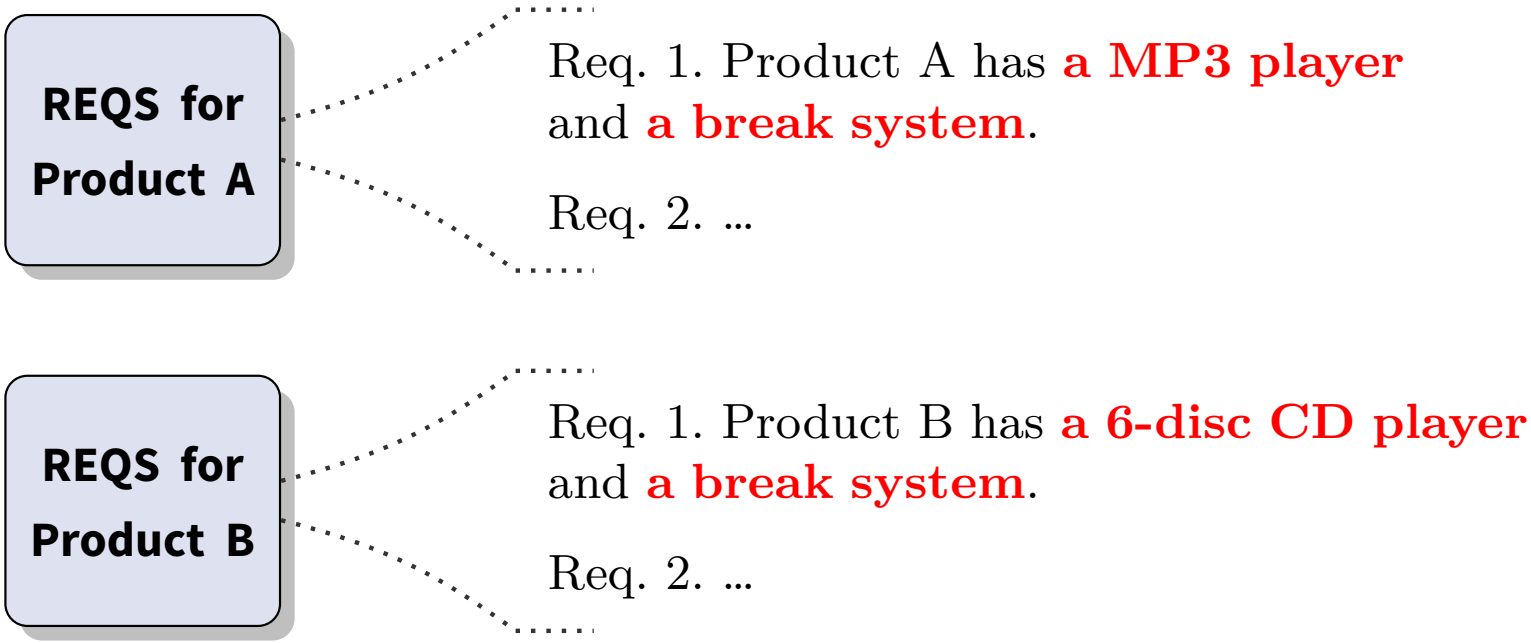
	Variability Abstraction	Abstraction-Realization Interface	Variability Realization
Variable Artifacts	Feature model	VPs and feature-to-VP mapping	Configurable platform *requirements, models, code, etc.*
Instances	Feature configs.	**VP configs.**	**Variants** ***requirements, models, code, etc.***

Variants (Models).



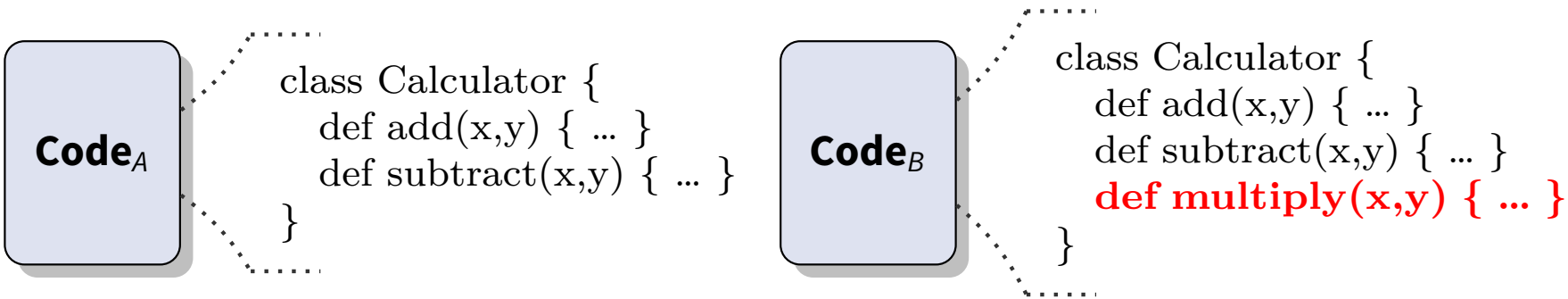
- (Ryssel et al. 2012) developed model matching on SimuLink models.
- (Rubin et al. 2012) developed model matching on UML state charts and class diagrams.

Variants (Requirements).



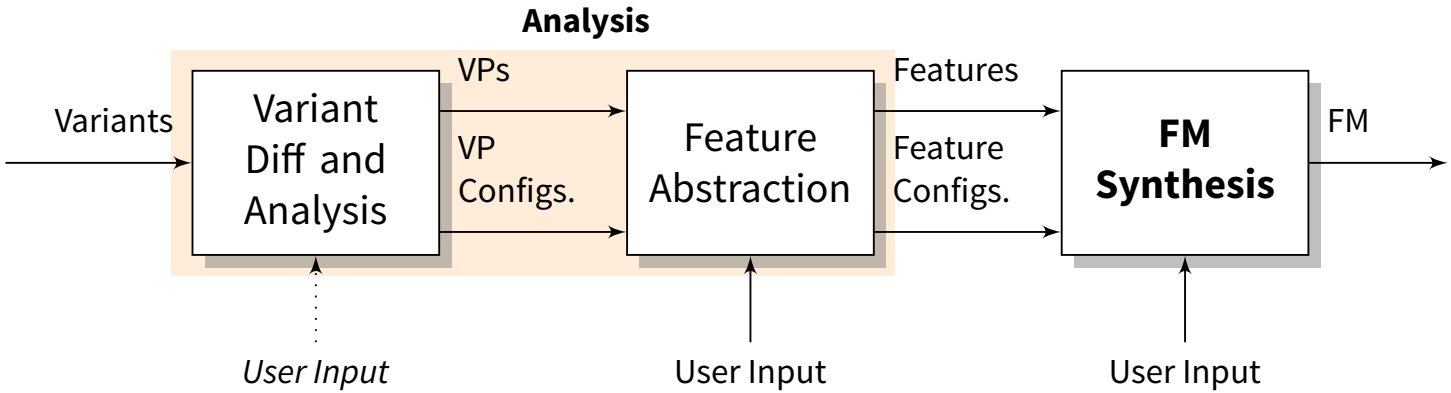
- Experience reported at an automotive company. VPs were manually identified between requirements documents.

Variants (Code).



- (Jepsen et al. 2007) report building a FM from products built with clone-and-own code variants at Danfoss drives.

Variant Workflows.



(Scenario. 2a, 2b, 2c) Variants (requirements, models, and code)

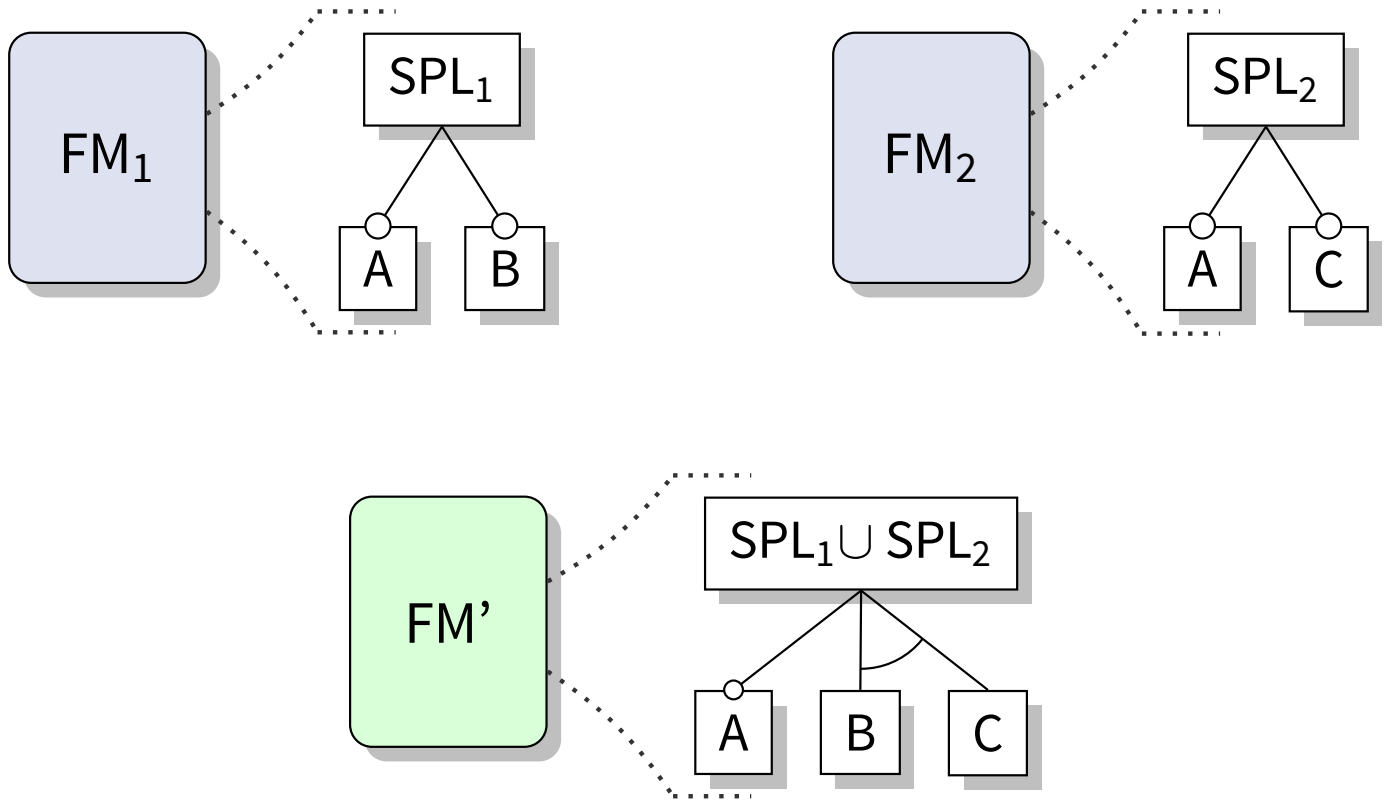
Additional scenario involving VP configs in the paper.

Scenario Outline.

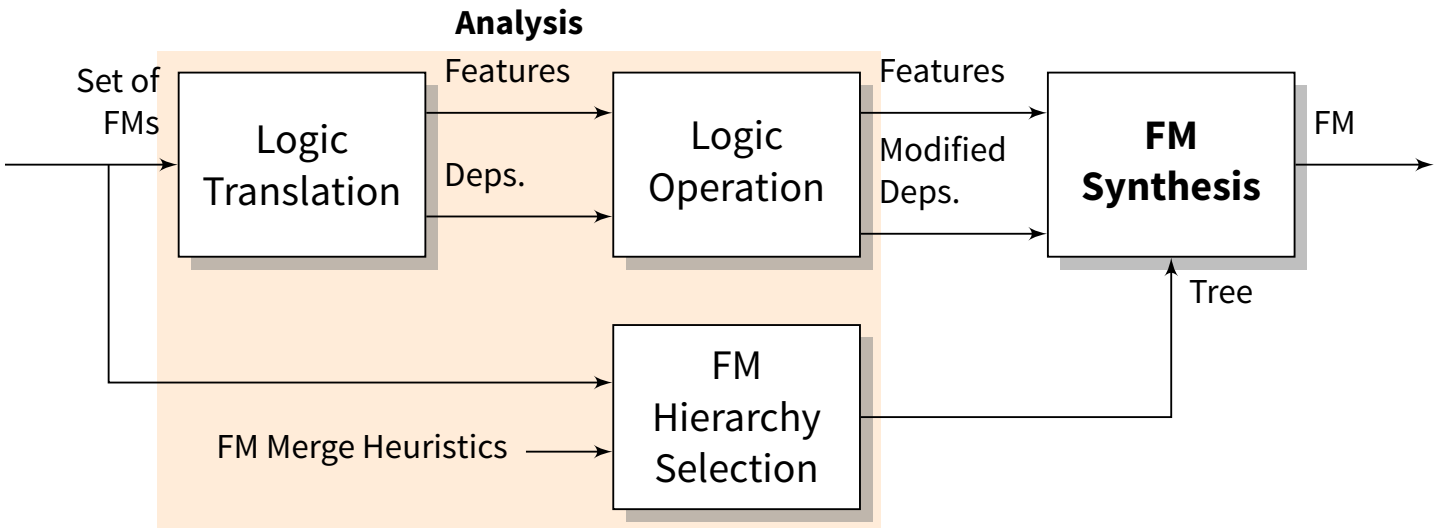
Scenario 3: Feature Model Operations

	Variability Abstraction	Abstraction-Realization Interface	Variability Realization
Variable Artifacts	**Feature model**	VPs and feature-to-VP mapping	Configurable platform *requirements, models, code, etc.*
Instances	Feature configs.	VP configs.	Variants *requirements, models, code, etc.*

Feature Model Operations.



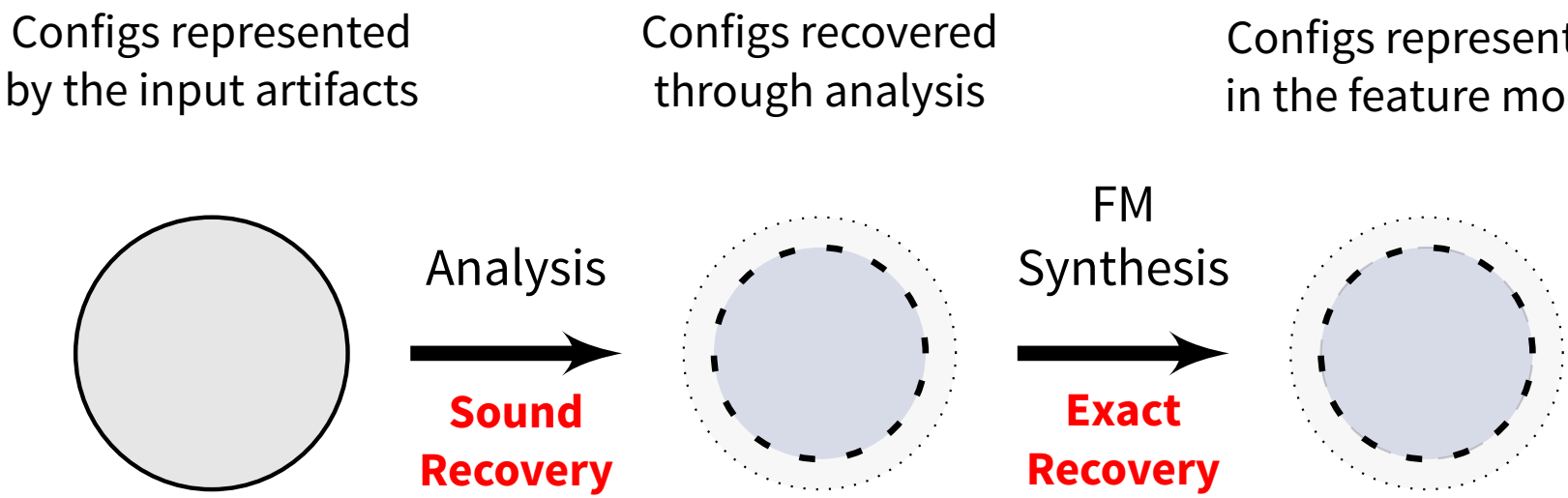
Feature Model Operations Workflow.



(Scenario. 3) Feature model operations

- (Acher 2011)'s thesis work was based on this scenario.
- (Fahrenberg et al. 2011) also described similar operations on models using FMs as an example.

Feature Model Operations: Precision.



Scenario Outline.

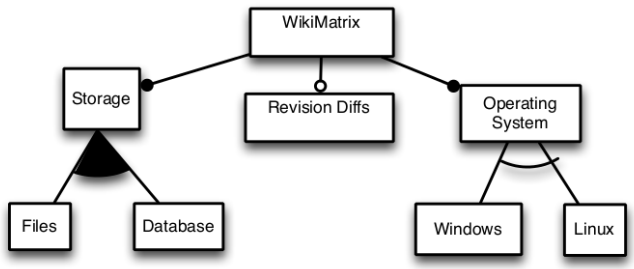
Scenario 4: Feature Model Merge Workflows

	Variability Abstraction	Abstraction-Realization Interface	Variability Realization
Variable Artifacts	**Feature model**	VPs and feature-to-VP mapping	Configurable platform *requirements, models, code, etc.*
Instances	Feature configs.	VP configs.	Variants *requirements, models, code, etc.*

Feature Model Merge Workflows.

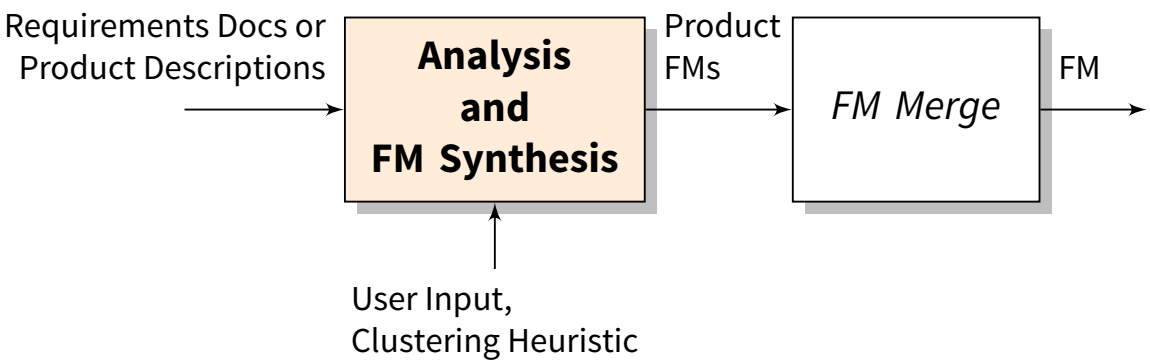
ID	Revision Diffs	Operating System	Storage	Open
...	...	...	...	...
FooWiki	Plugin	Windows; Linux	Files ; Database	No
...	...	...	...	...

(a) A wiki engine description



(b) Corresponding FM

Product descriptions (Acher et al. 2012)



Outline.

Discussion.

- Analysis of *variable artifacts* recover **dependencies**, while variants recover **sets of VP or feature configurations**.
- Alternatively, **FMs** can be used as input (e.g. FM operations and merge).
- **Heuristics** or **user input** can be used to select a distinct hierarchy for the FM depending on the scenario.
- Additional configurations recovered by a **complete** analysis could be pruned with a **sound** synthesis and vice versa.

Conclusions.

- FM synthesis is required in a wide range of scenarios with significantly different input artifacts.
- Different input artifacts require different analysis techniques and synthesis workflows.
- Scenarios can be used for a qualitative evaluation of FM synthesis techniques.
- Usage scenarios provide requirements for synthesis techniques.