KDM Analytics™

High-Fidelity analysis of software systems

Dr. Nikolai Mansourov
Chief Technology Officer, KDM Analytics

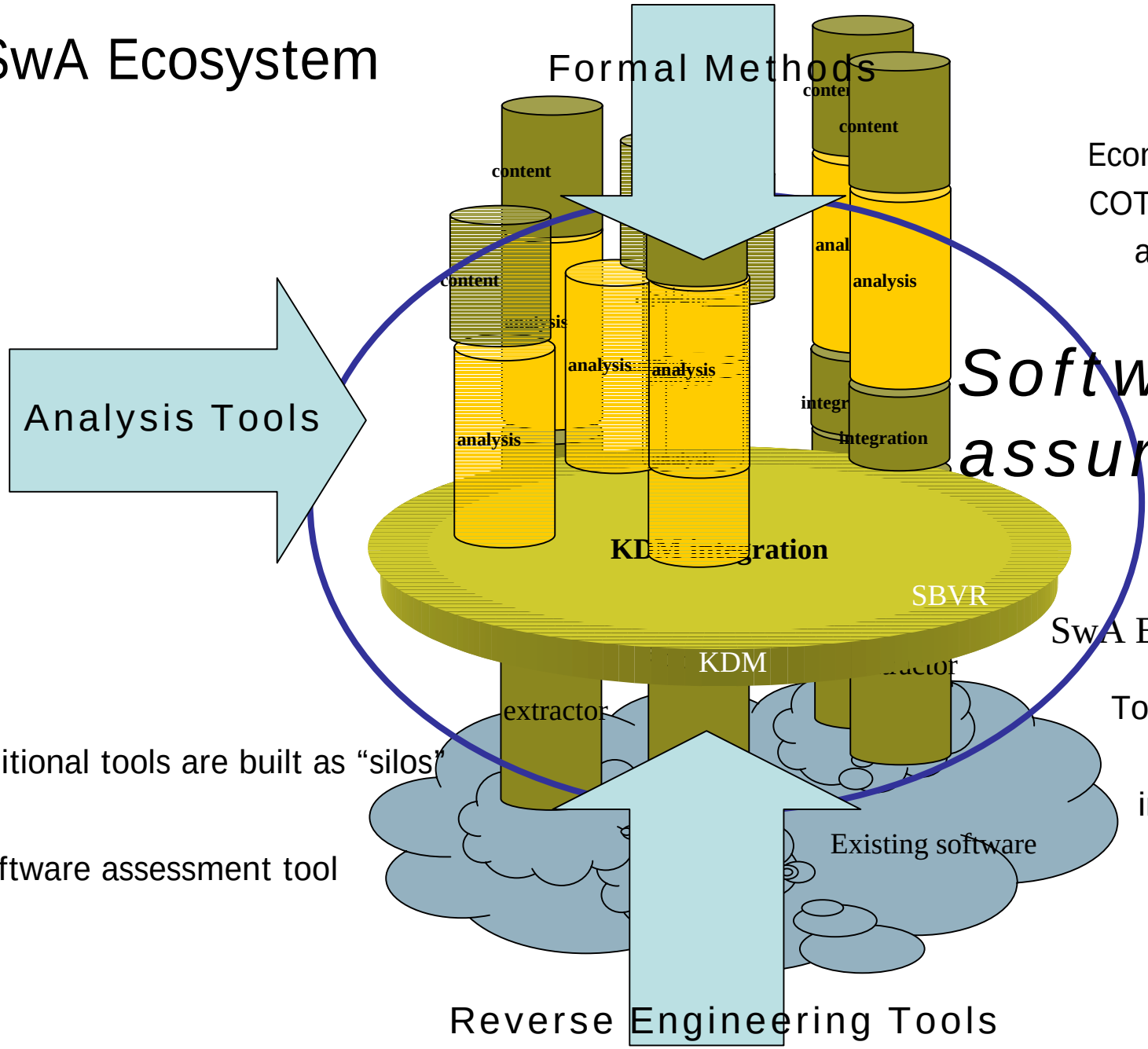
http://www.kdmanalytics.com

5 March 2007

Agenda

1. Motivation: analysis of security properties of existing software systems
2. The SwA Ecosystem
3. OMG Knowledge Discovery Metamodel (KDM)
4. Fidelity of the static analysis
5. Micro KDM – high fidelity intermediate representation
6. Platform knowledge in KDM
7. Next steps

SwA Ecosystem



Economy of tools, COTS components and services emerges

SwA Ecosystem tools

Today's tools have limited interoperability

Traditional tools are built as "silos"

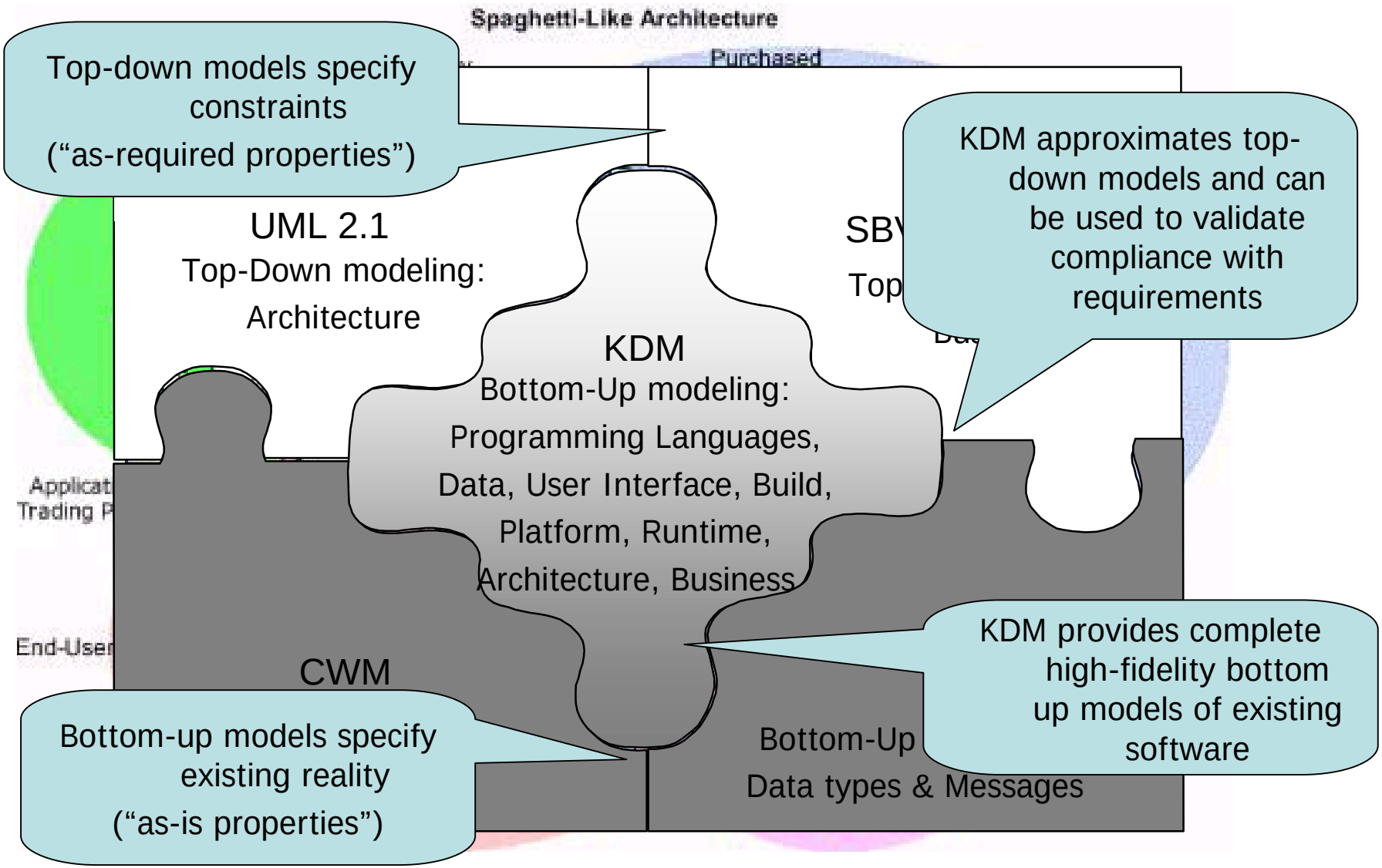
software assessment tool

Why KDM ?

KDM is an OMG *foundation* for software assurance and modernization

- Meta-model for describing existing software assets
  - Includes software, its parts and operating environment
  - Focuses on structure of existing software to provide context for modernizations
- Represents existing software as Entities, Relationships and Attributes
- Exchange of information between tool vendors
- Language- and platform-independent yet extensible
- Spans multiple abstractions yet traceable back to the artifacts

KDM: precise models of existing enterprise software



© Gartner Group

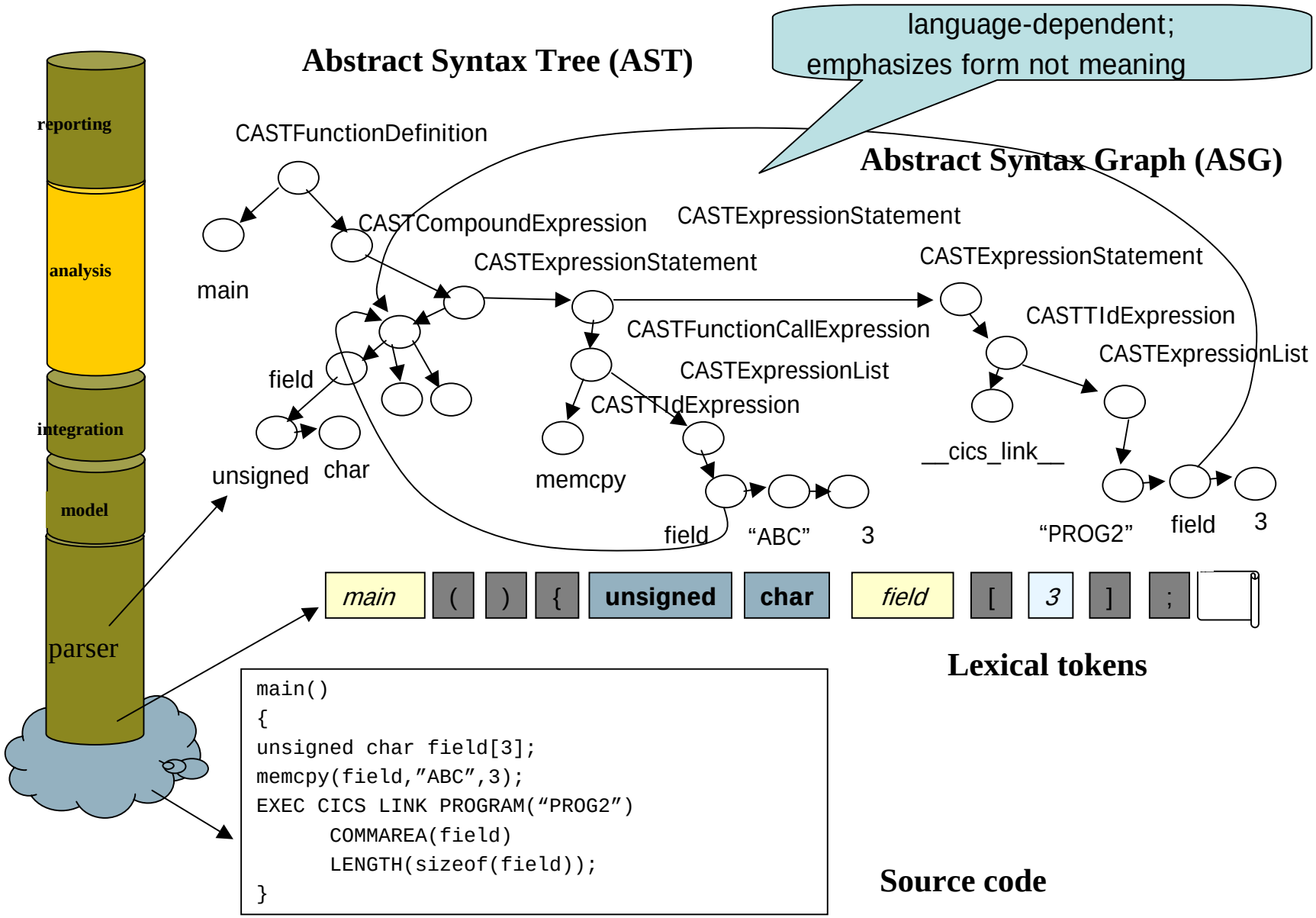
Motivation: High-fidelity static analysis

- Fidelity: trustfulness, closeness in sound, facts, color, etc. to the original
  - Fidelity of the internal model
  - Accuracy
  - Adequacy
  - Proper scope

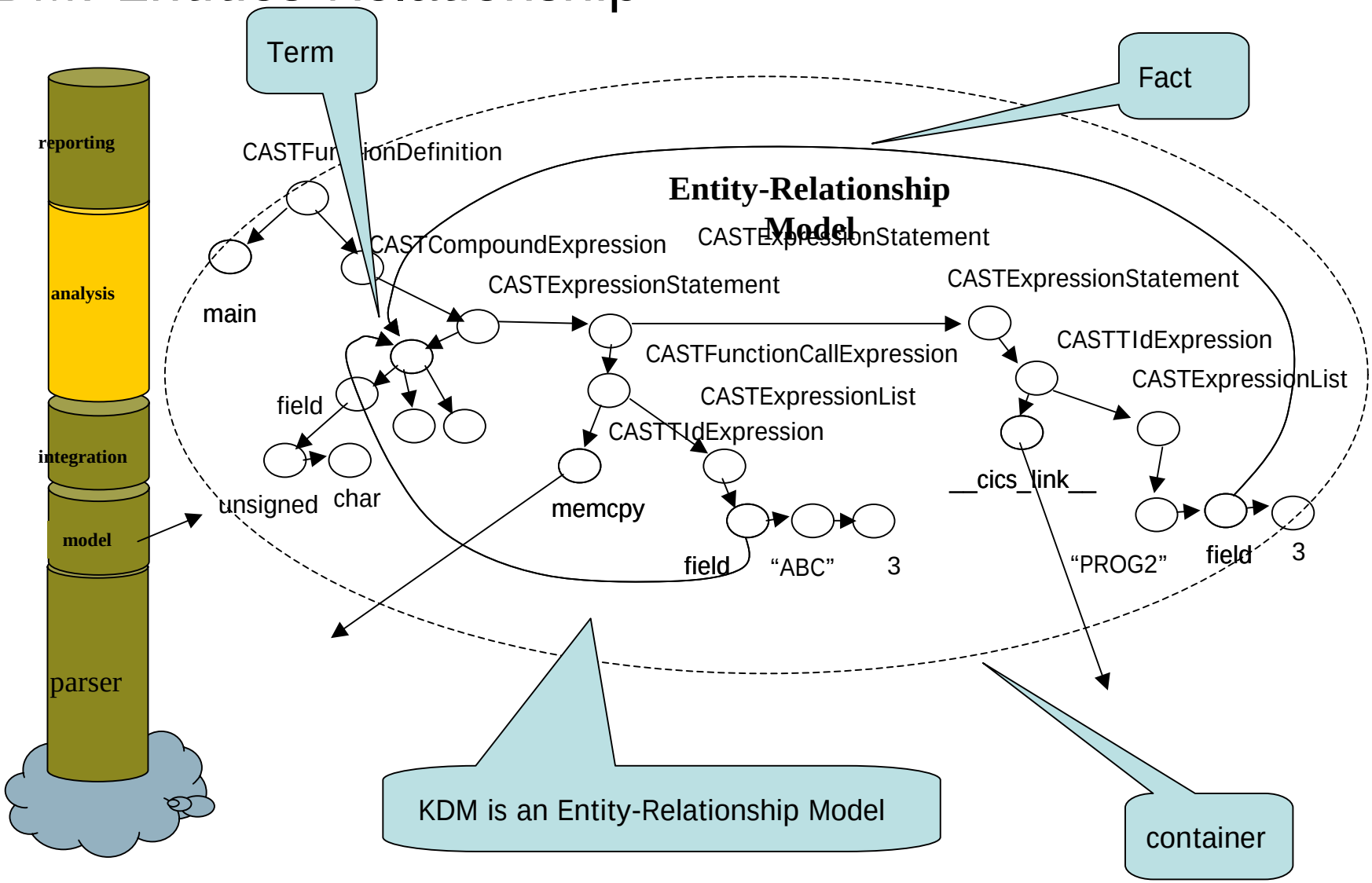
Factors that Limit Fidelity

- Low fidelity models
- Technical platform considerations (even a high-fidelity model may be incomplete)
- Other mismatches
  - Dynamic aspects
  - Preprocessor
  - Code generation

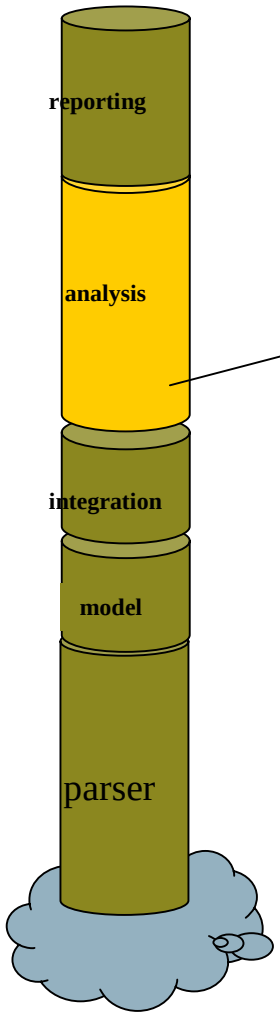
Background: Language-dependent models

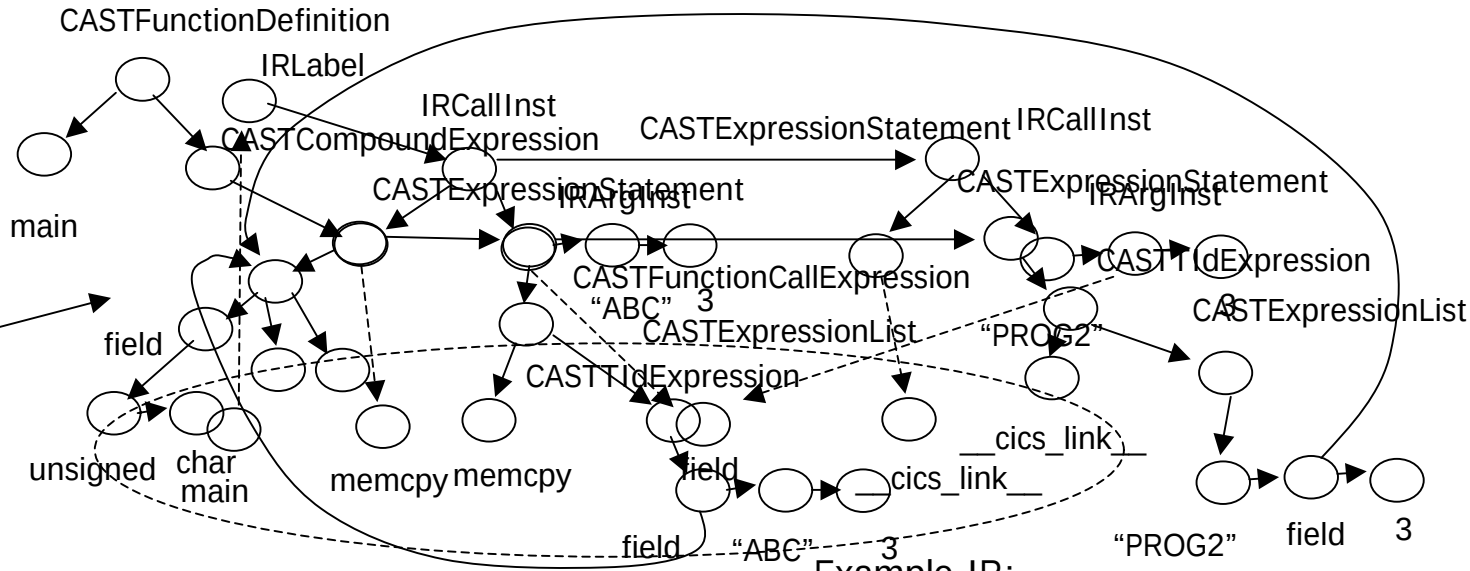


KDM: Entities-Relationship



Micro KDM: Lower-Level models of software





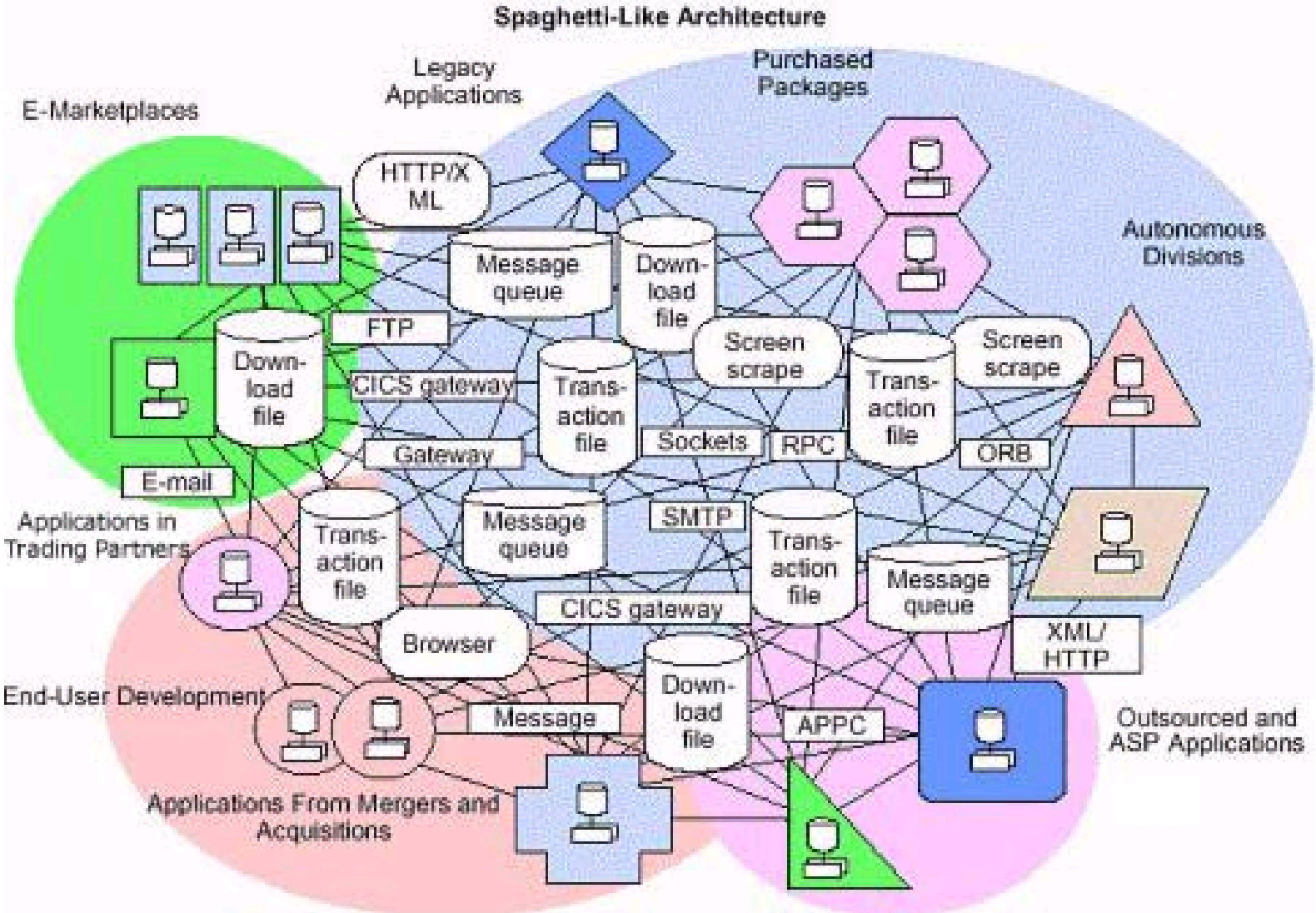
Intermediate Representation (IR)

KDM terms and facts reach down to this level

Example IR:

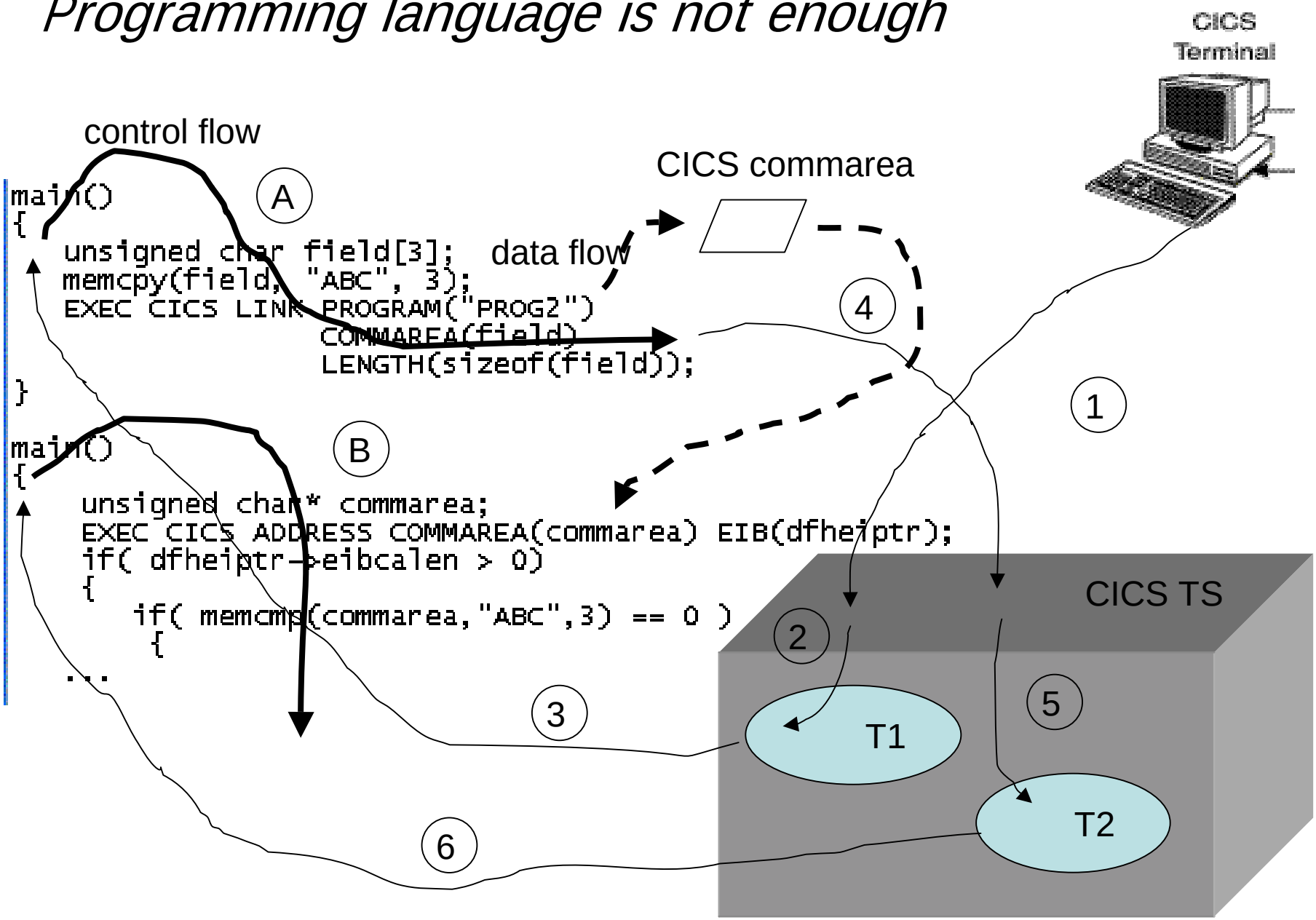
- Sun Compiler 3-address IR
- Intel compiler XIL
- DEC GEM Compiler for Alpha
- Java bytecode
- Microsoft CLR
- DIANA IR for Ada
- SUIF compiler
- RTL of GCC compiler
- GIMPLE of GCC compiler
- LAST of the McCAT compiler

Evaluation target – enterprise software system

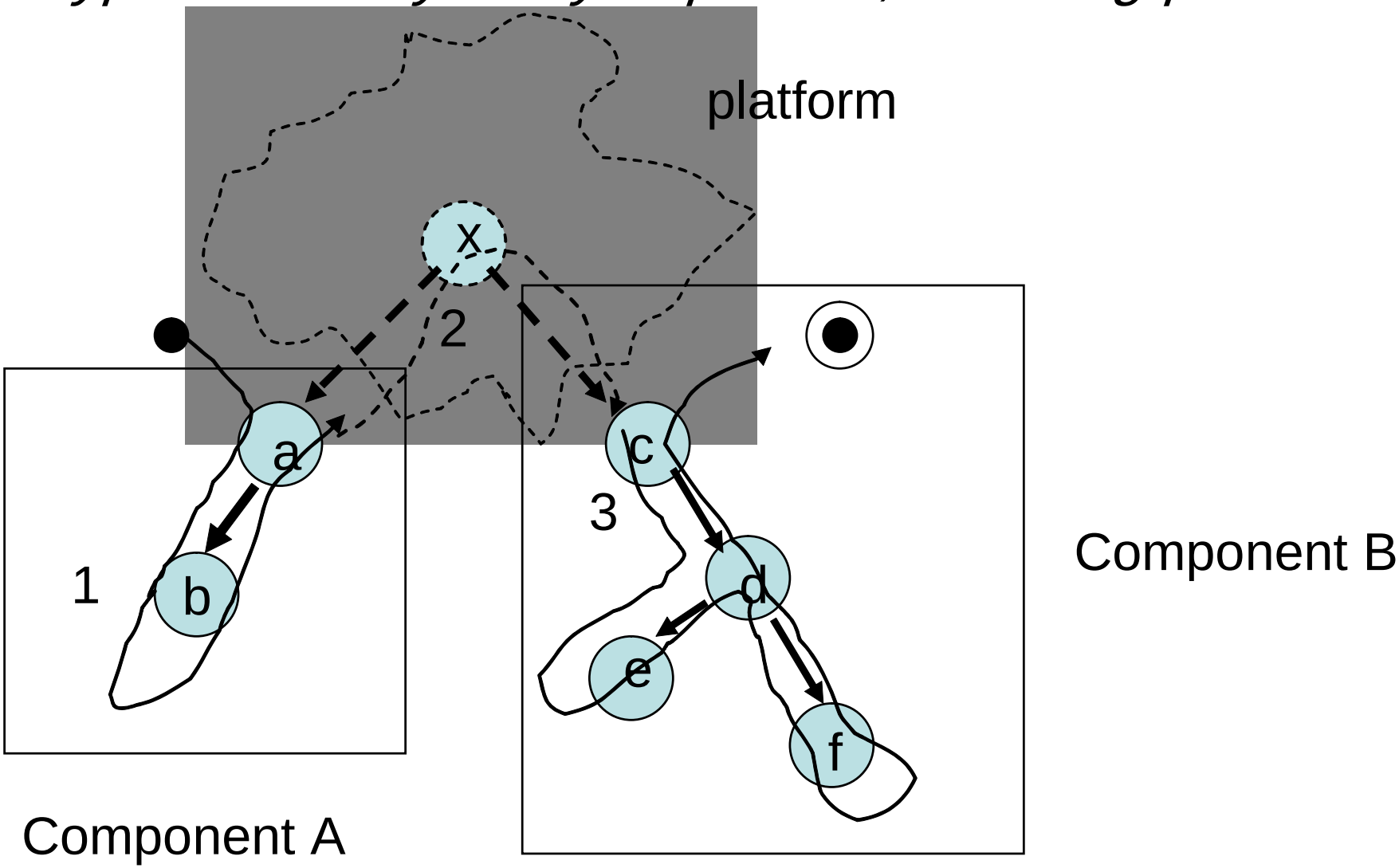


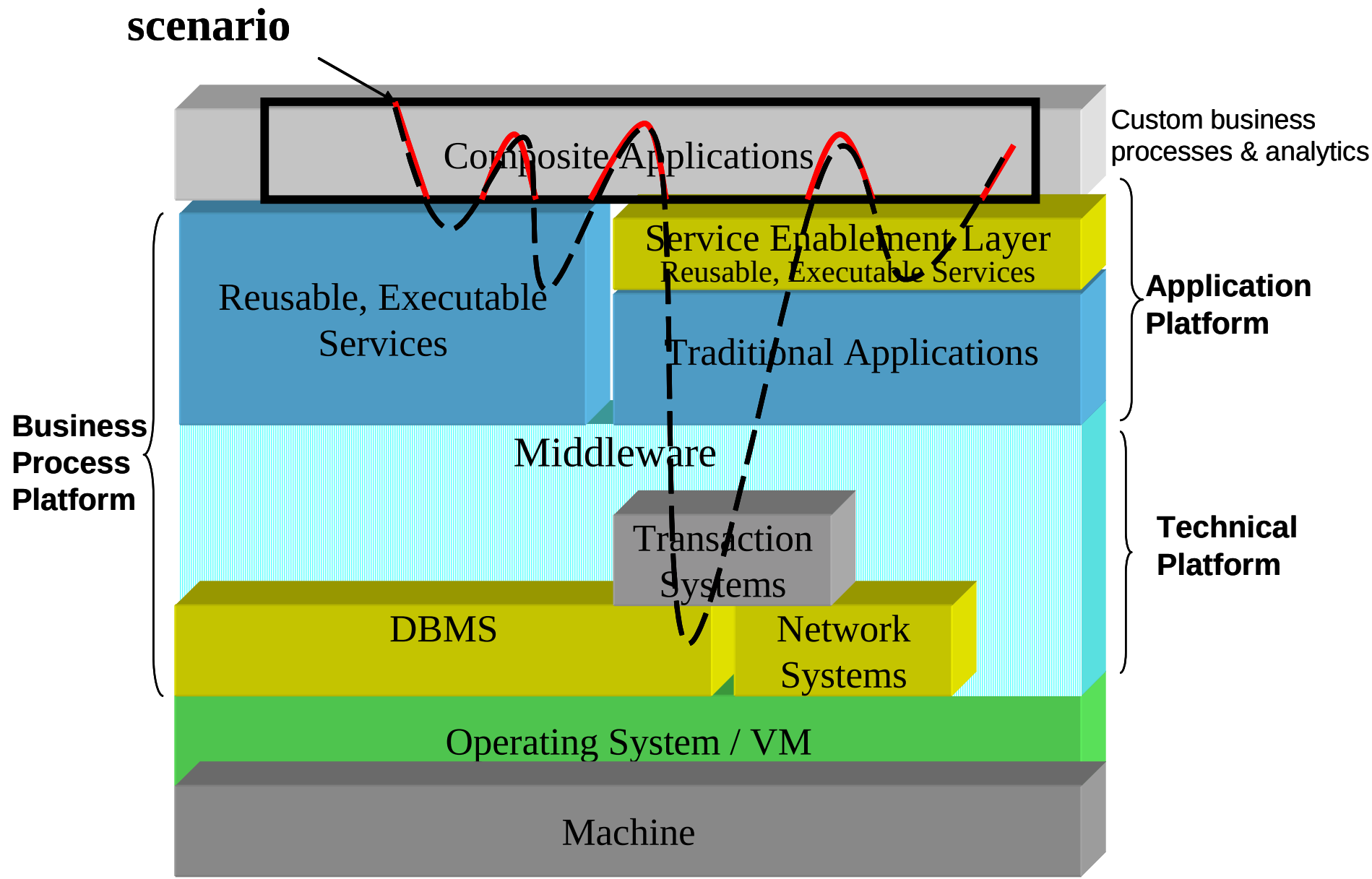
© Gartner Group

Programming language is not enough



Typical security analysis pattern, involving platform

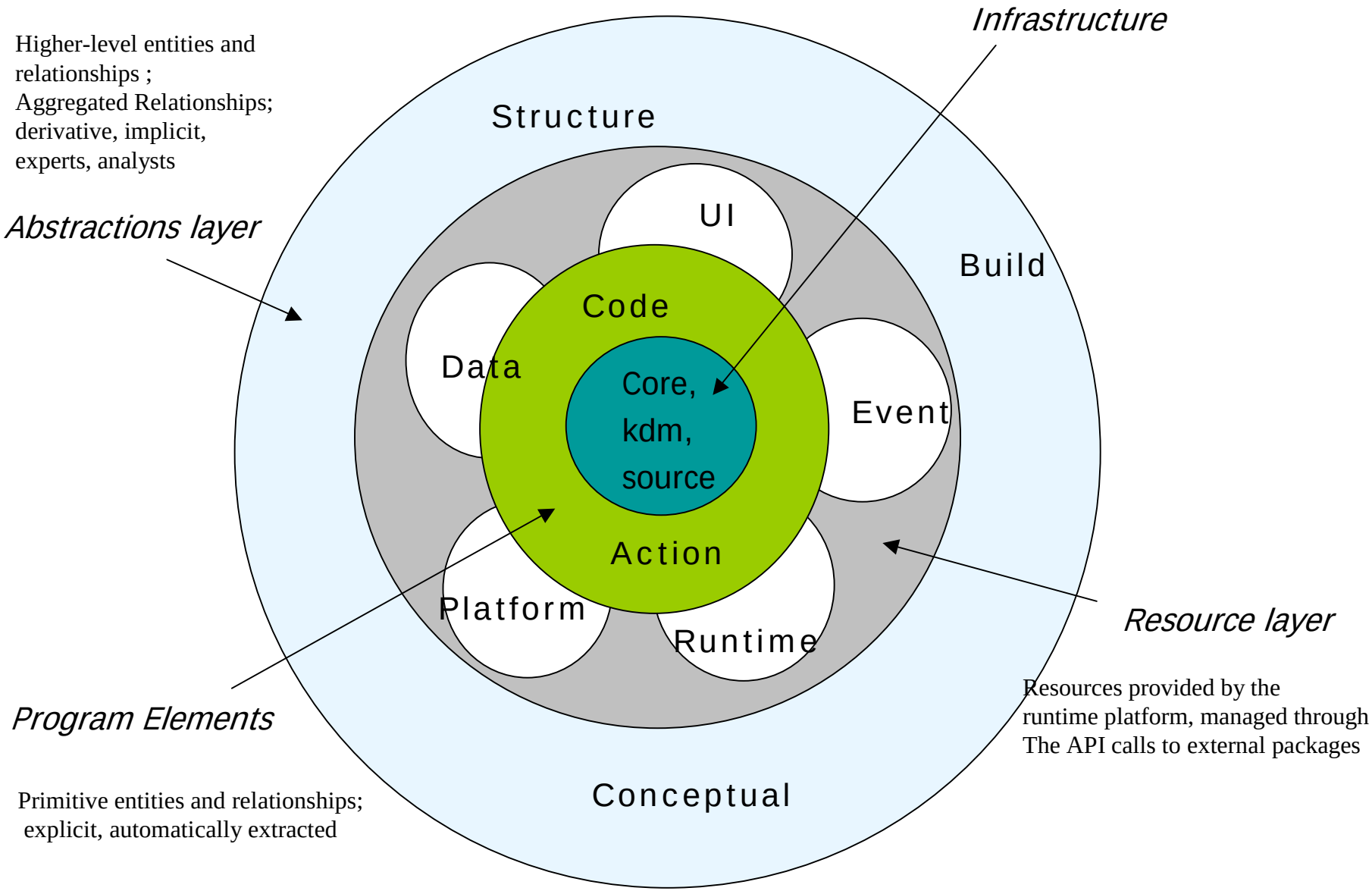




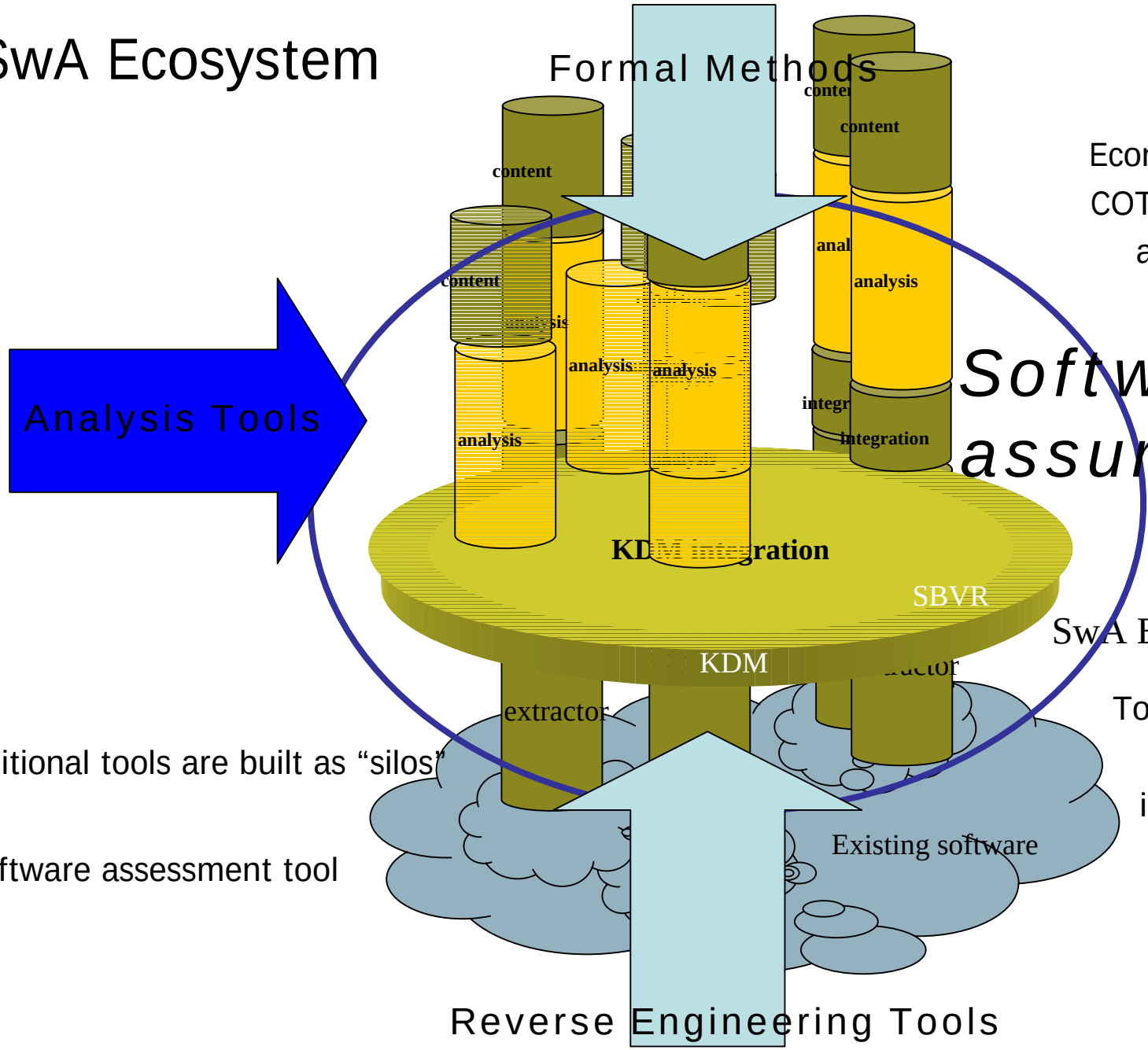
Platform knowledge

- platform provides *resources* to application code
- platform provides *services* that are related to resources
- application code invokes platform services to *manage the life-cycle* of resources
- platform provides *component deployment mechanism*
- platform defines *control and data flow* between application components
- platform provides *error handling* across application components
- platform provides *integration* of application components

KDM architecture



SwA Ecosystem



Economy of tools, COTS components and services emerges

SwA Ecosystem tools

Today's tools have limited interoperability

Traditional tools are built as "silos"

software assessment tool

Thank you!