Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Multiple Runtime Models and their Relations for Self-Managing Systems

HPI/SAP Workshop

Walldorf, Germany, December 06, 2010

Thomas Vogel

HPI Research School

System Analysis & Modeling Group

Motivation

- **Continuous adaptation** of software to keep its value for the user (Laws of Software Evolution) [Lehman, 1996]
- (Increasing) **complexity** of software systems [Northrop et al., 2006]
- Maintenance & administration costs [Sterritt, 2005, Sommerville, 2007]

Motivation

- **Continuous adaptation** of software to keep its value for the user (Laws of Software Evolution) [Lehman, 1996]
- (Increasing) **complexity** of software systems [Northrop et al., 2006]
- Maintenance & administration costs [Sterritt, 2005, Sommerville, 2007]

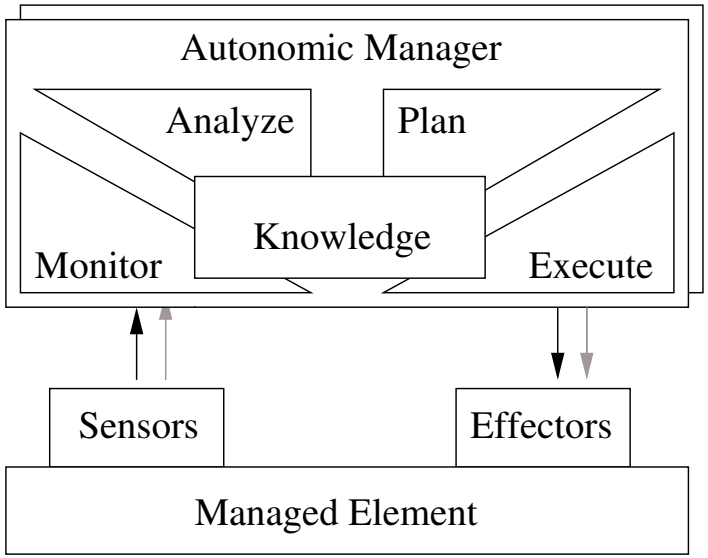
Self-Adaptive/-Managing Software [Cheng et al., 2009]

Systems that are able to adjust their behavior in response to their perception of the environment and the system itself.

$\rightsquigarrow$ Autonomic Computing

[Kephart and Chess, 2003]

Self-Managing Systems



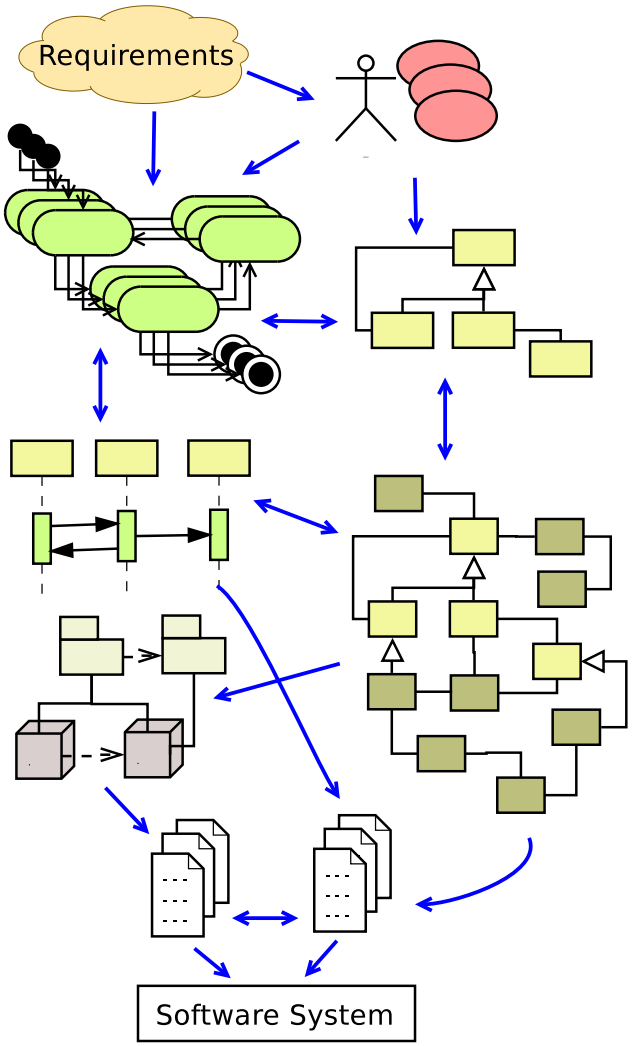
Feedback Loop [Kephart and Chess, 2003]

- Concepts originating from the control engineering discipline
  [Kokar et al., 1999, Diao et al., 2005]
- Self-healing/-optimization/-protection/-configuration [Lin et al., 2005]

Model-Driven Engineering

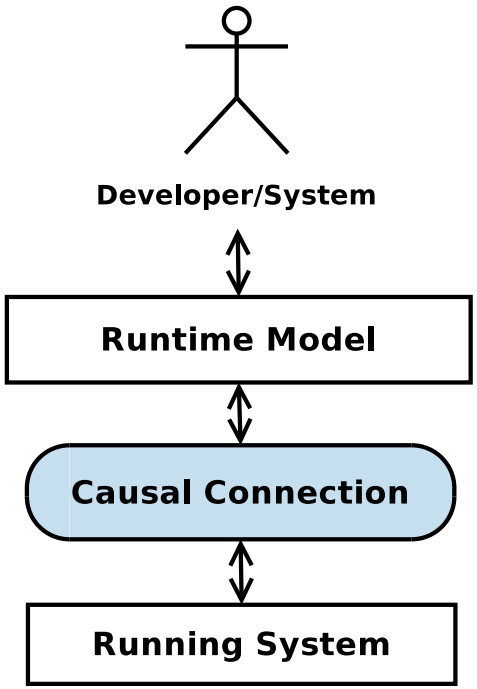
*The term Model-Driven Engineering (MDE) is **typically** used to describe **software development** approaches in which abstract models of software systems are created and systematically transformed to concrete implementations.*

[France and Rumpe, 2007]



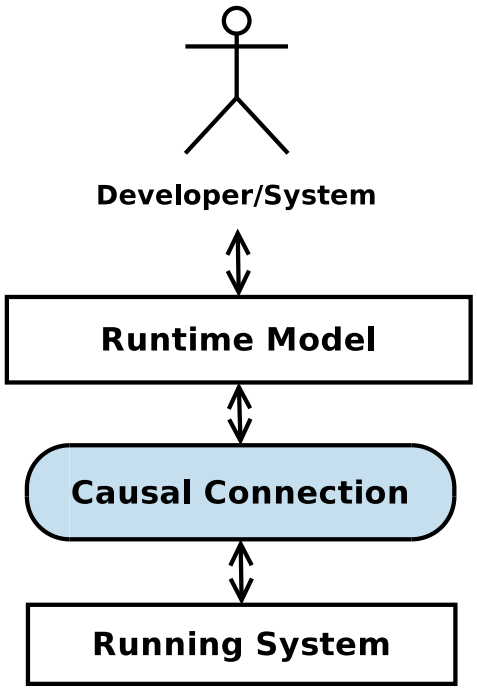
MDE & Self-Managing Systems

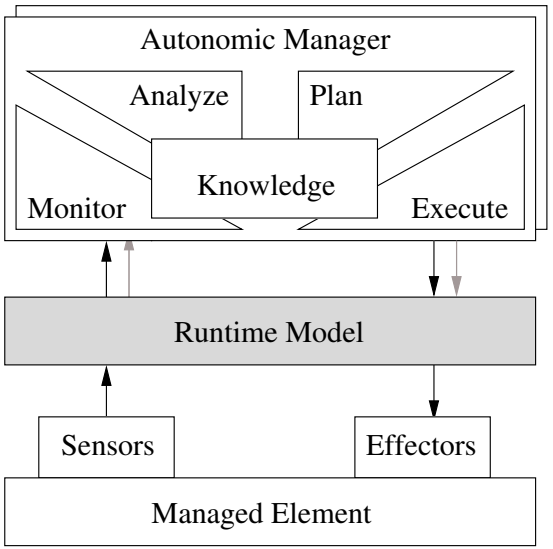
- *In our broad vision of MDE, models [. . .] are also the **primary means by which developers and other systems understand, interact with, configure and modify the runtime behavior of software.*** [France and Rumpe, 2007]
- *Models@run.time* [Blair et al., 2009]



MDE & Self-Managing Systems

- *In our broad vision of MDE, models [. . .] are also the **primary means by which developers and other systems understand, interact with, configure and modify the runtime behavior of software.*** [France and Rumpe, 2007]
- *Models@run.time* [Blair et al., 2009]

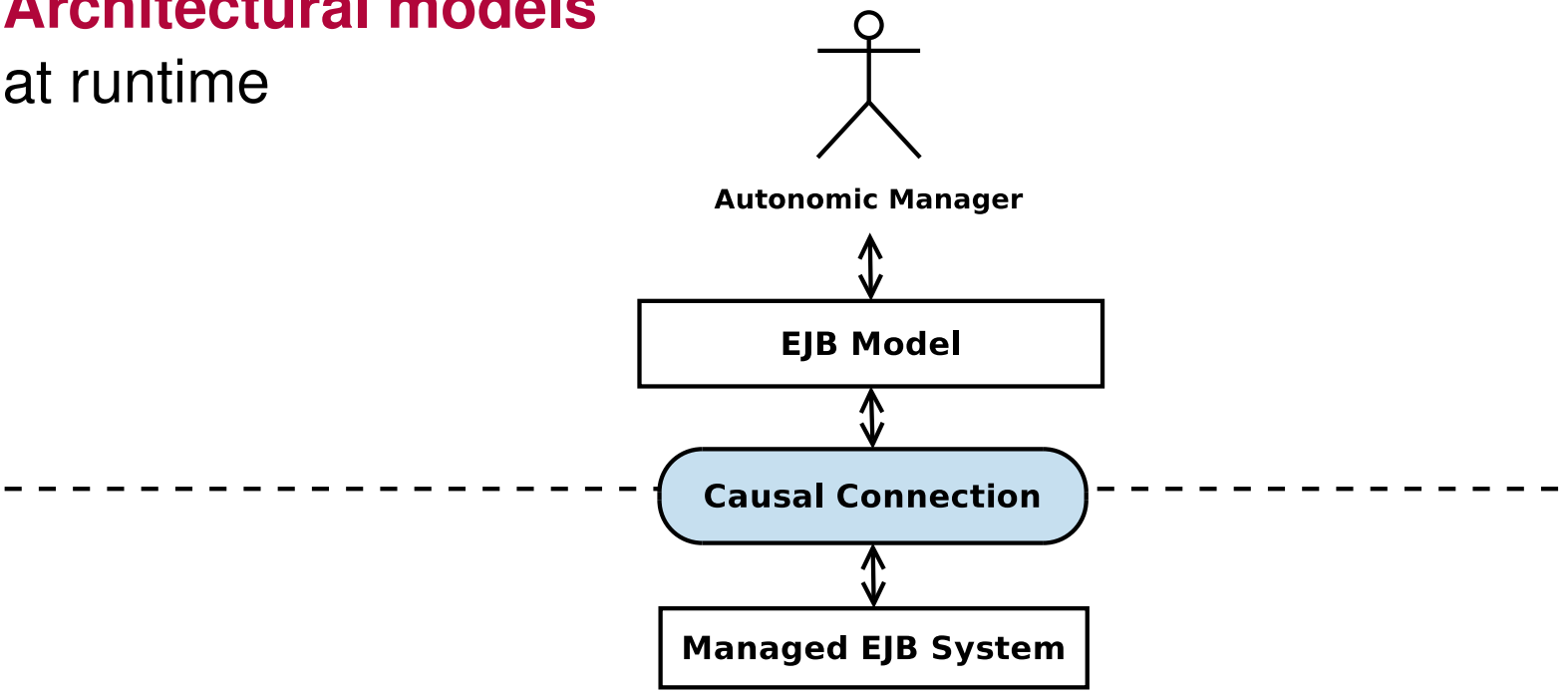




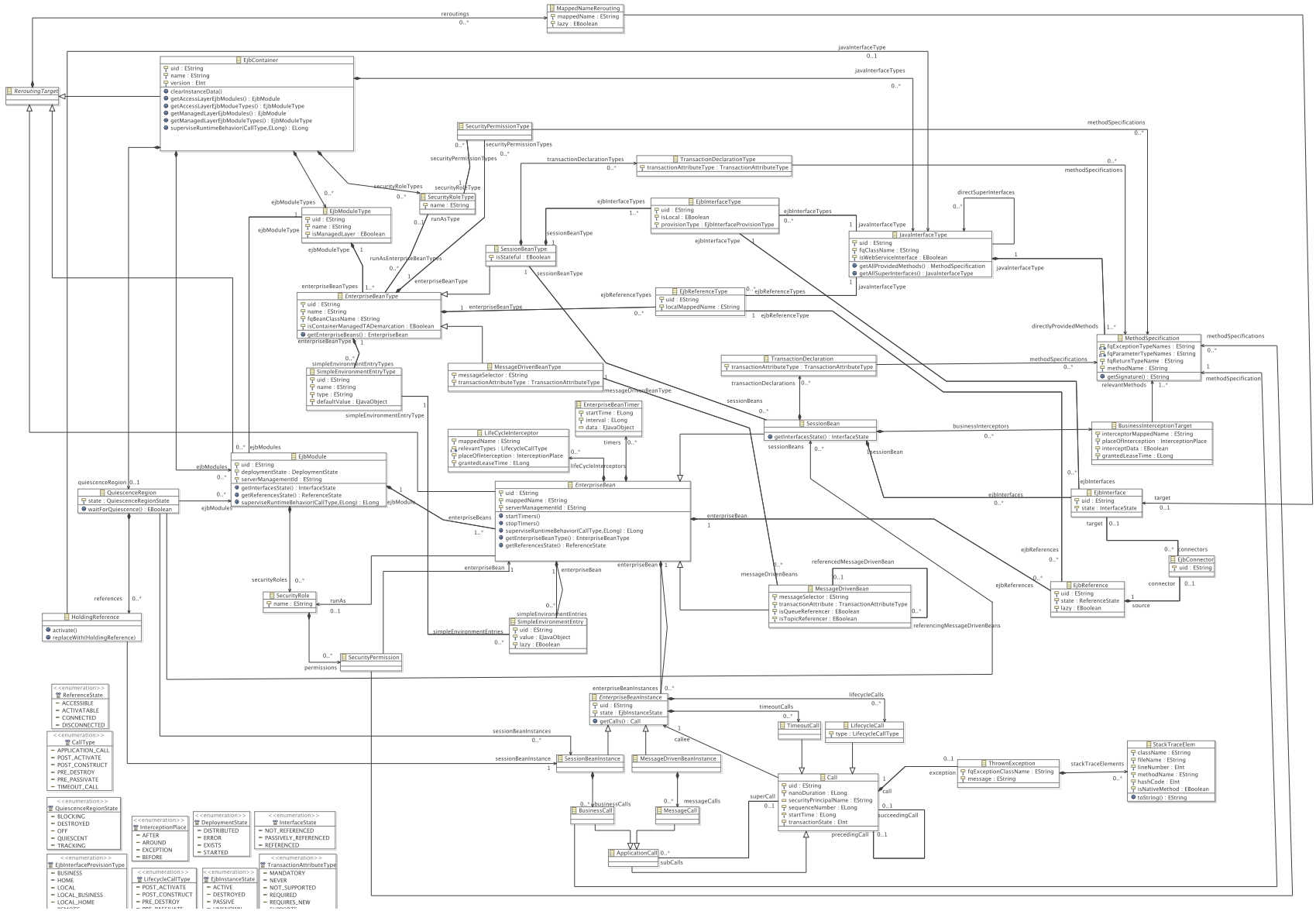
Self-Managing Systems

Self-Managing EJB Systems

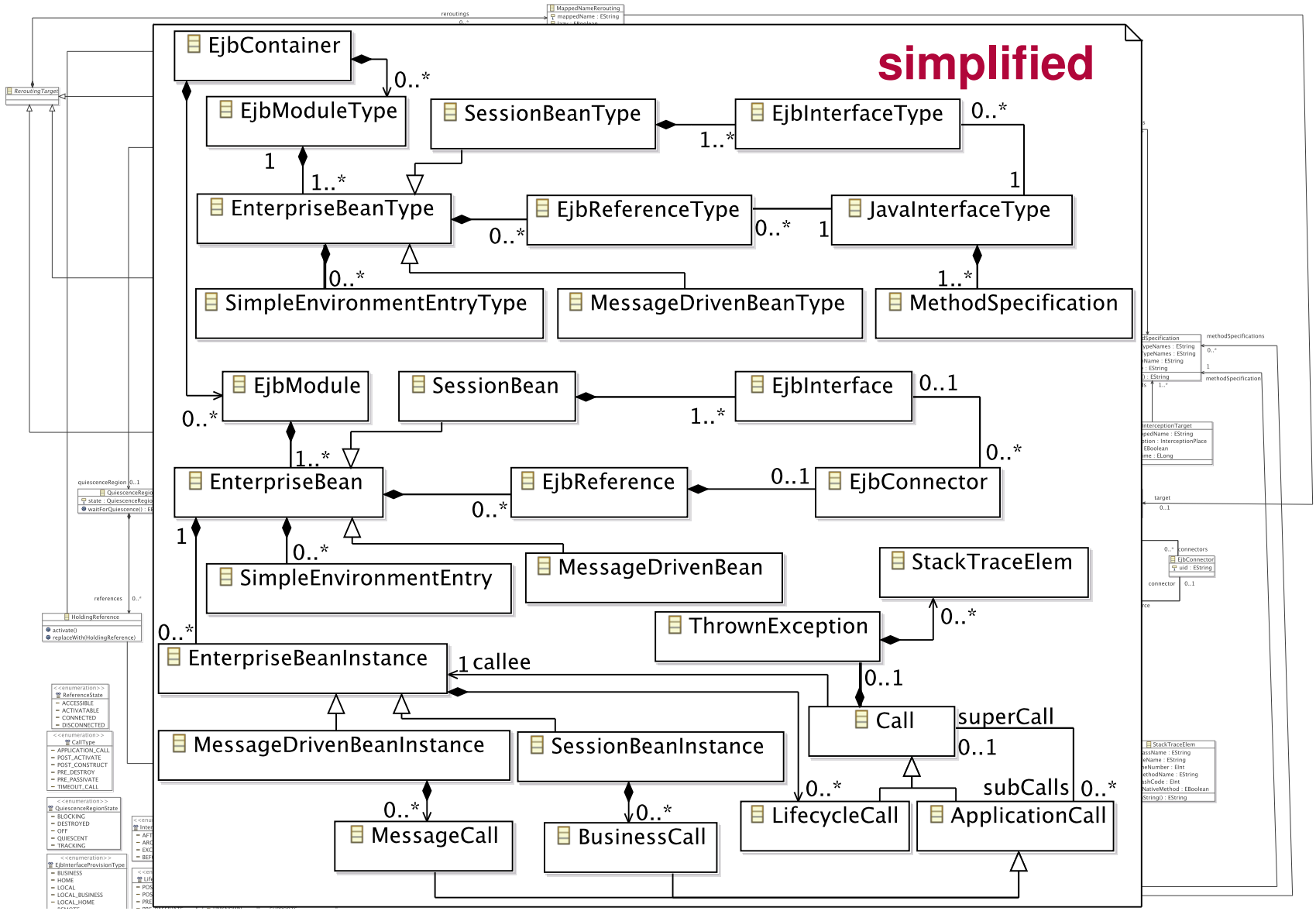
- Component-based software systems
  - *Enterprise Java Beans 3* (EJB)
- **Architectural monitoring and adaptation**
  - components and connectors, above the level of *Java* objects
- **Architectural models**
  at runtime



EJB (Runtime) Metamodel

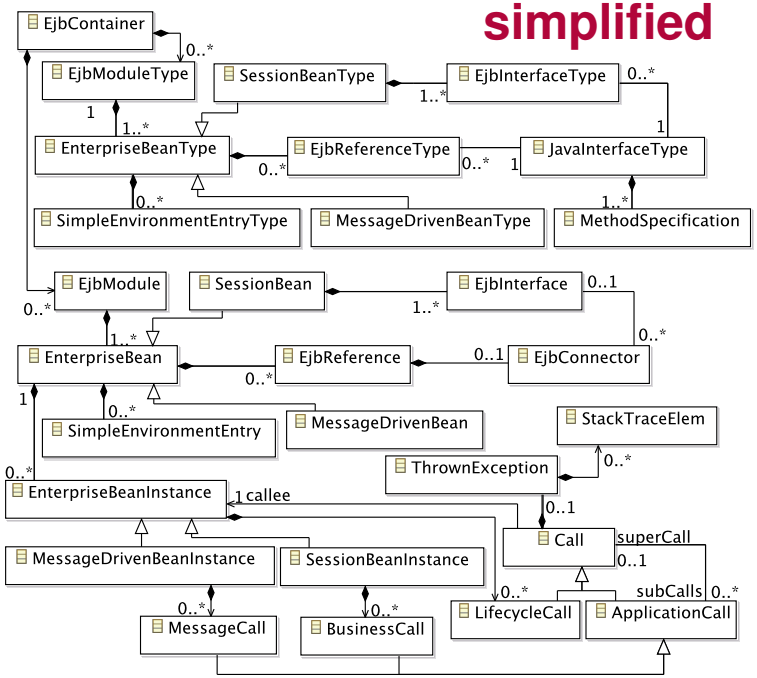


EJB (Runtime) Metamodel



Abstract Runtime Metamodels

simplified



complex

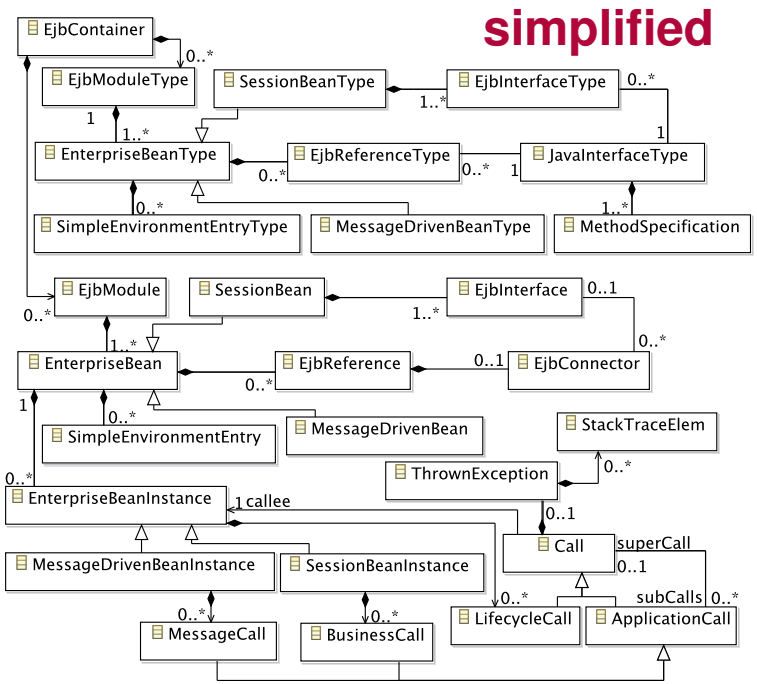
detailed

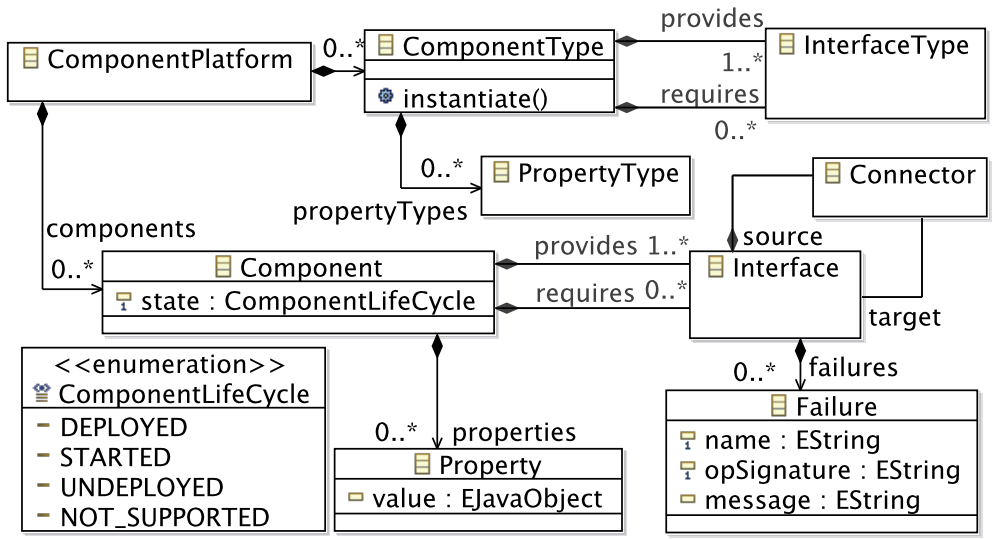
multiple concerns

platform-specific

solution space

Abstract Runtime Metamodels



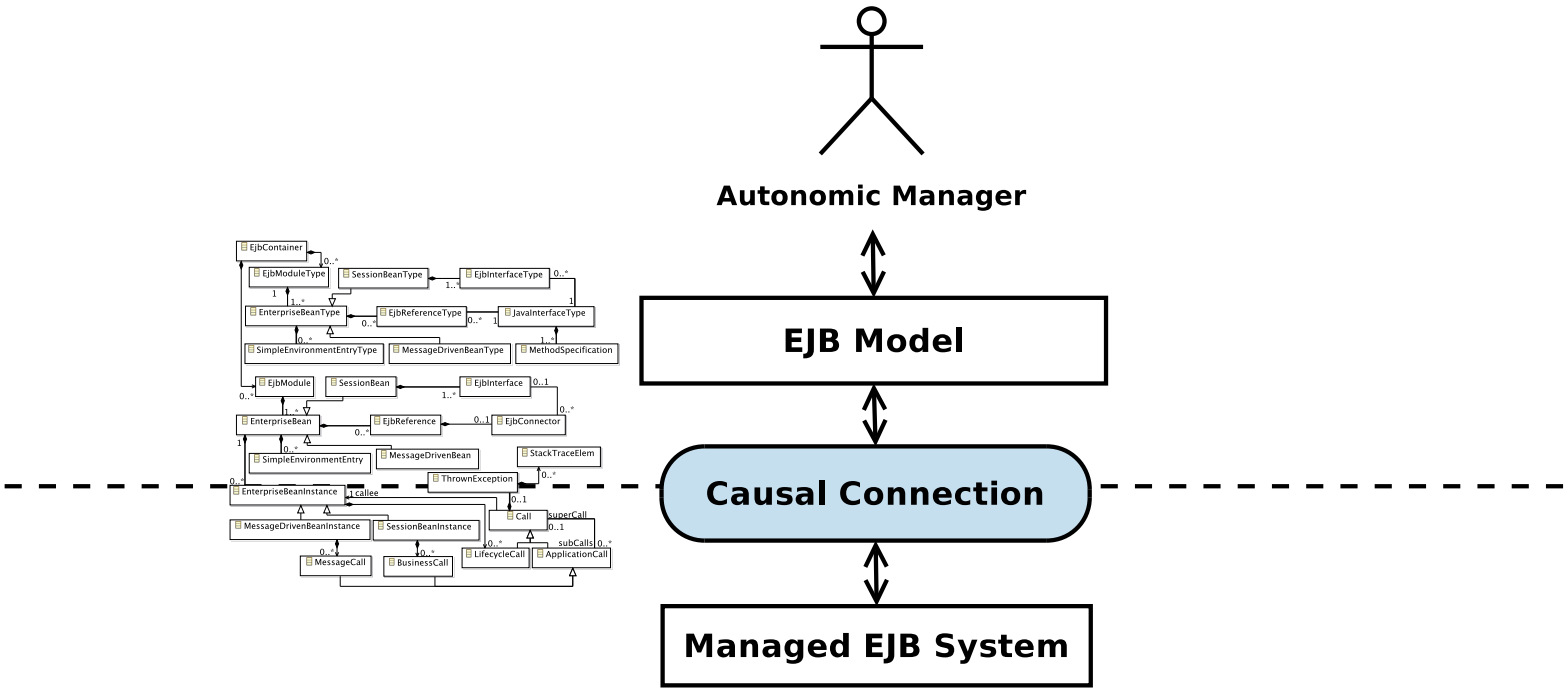


complex
detailed
multiple concerns
platform-specific
solution space

vs.

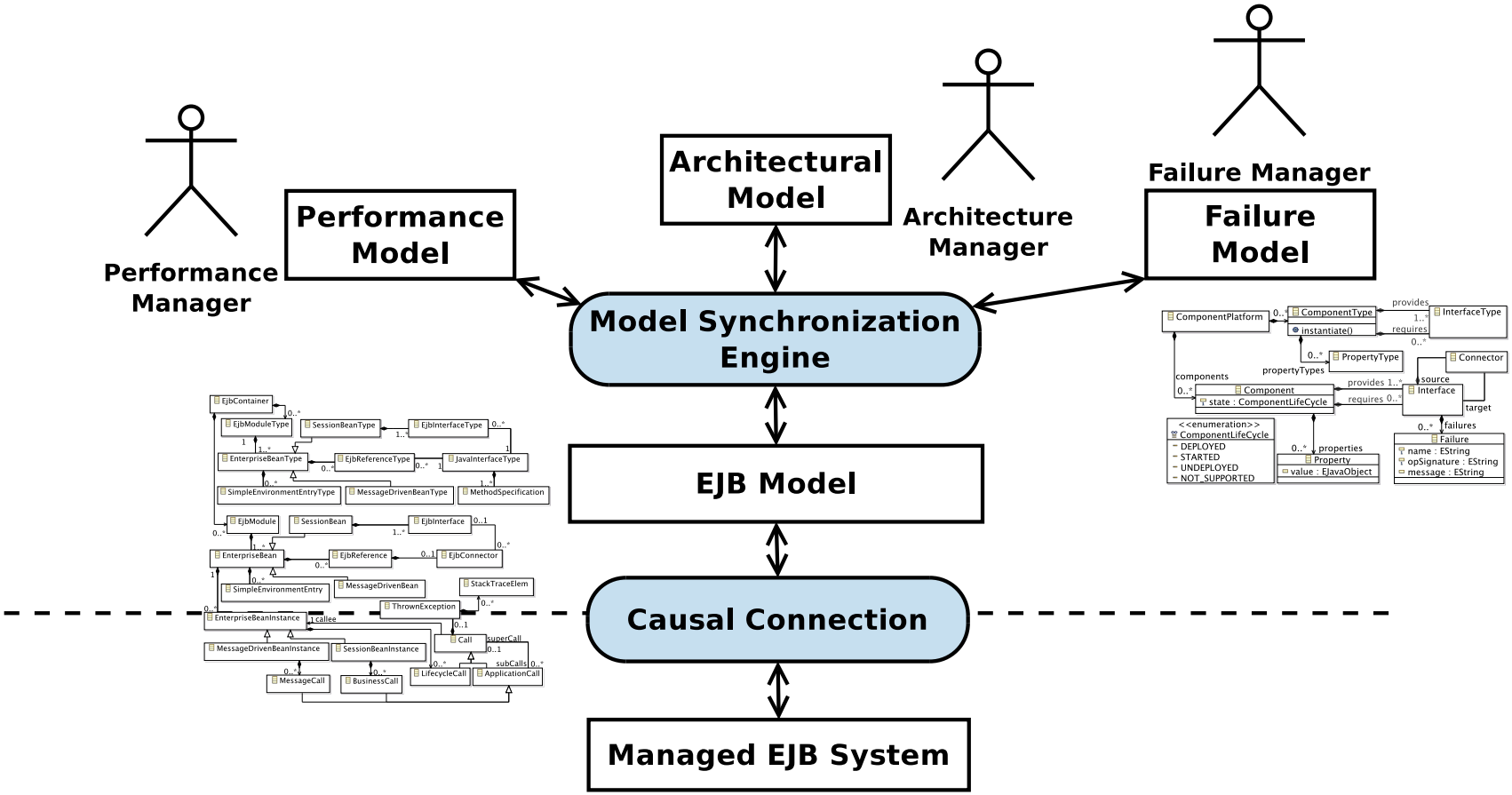
less complex
abstract
one concern
platform-independent
problem space

Abstract Runtime Models



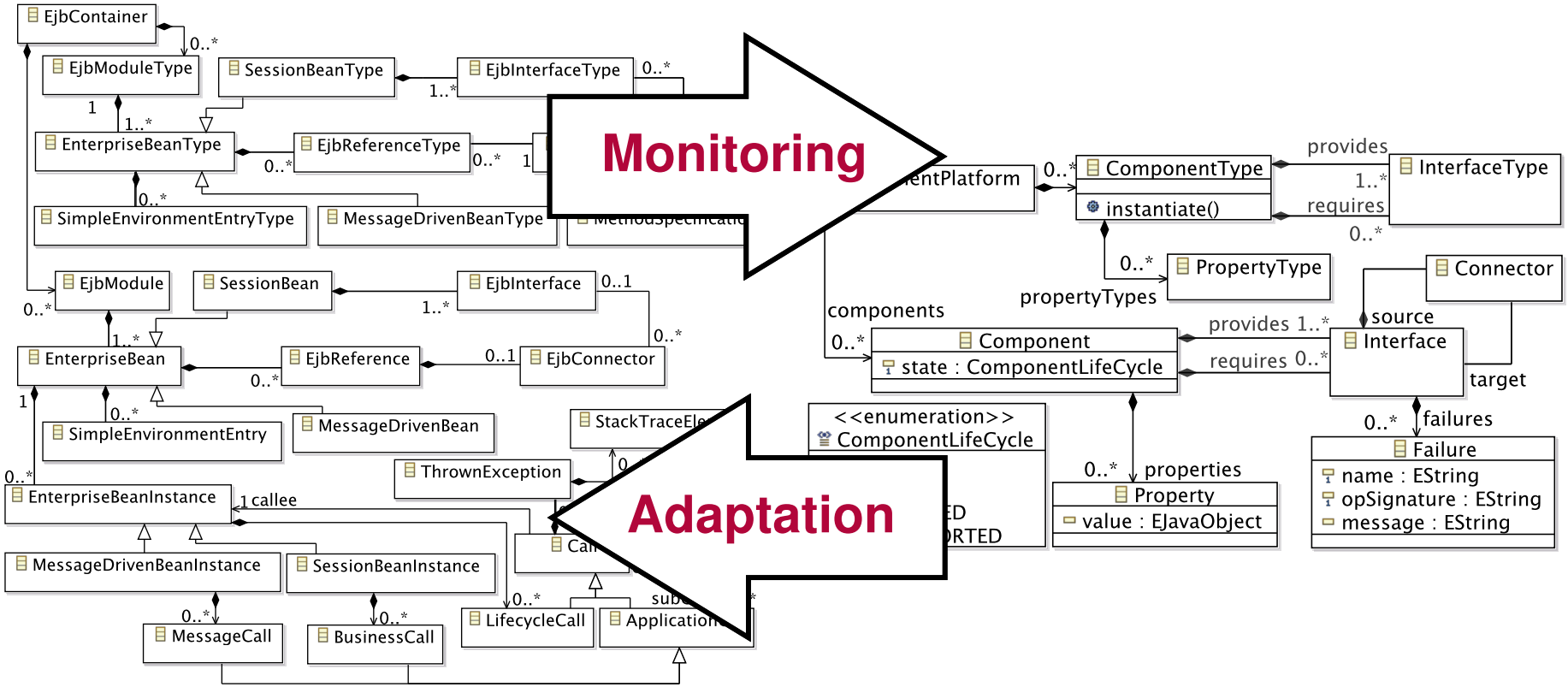
Abstract Runtime Models

- Concern-specific, platform-independent models and managers
- **Model Synchronization** to maintain the runtime models



Runtime Model Synchronization

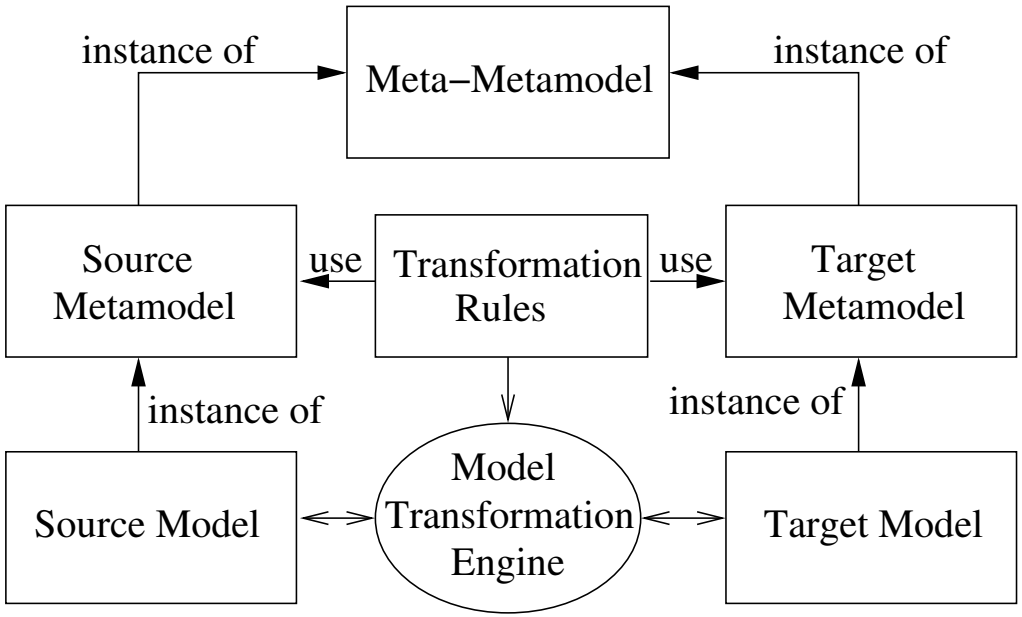
- Abstraction (monitoring) and refinement (adaptation)



Simplified **EJB Metamodel**
(Source Metamodel)

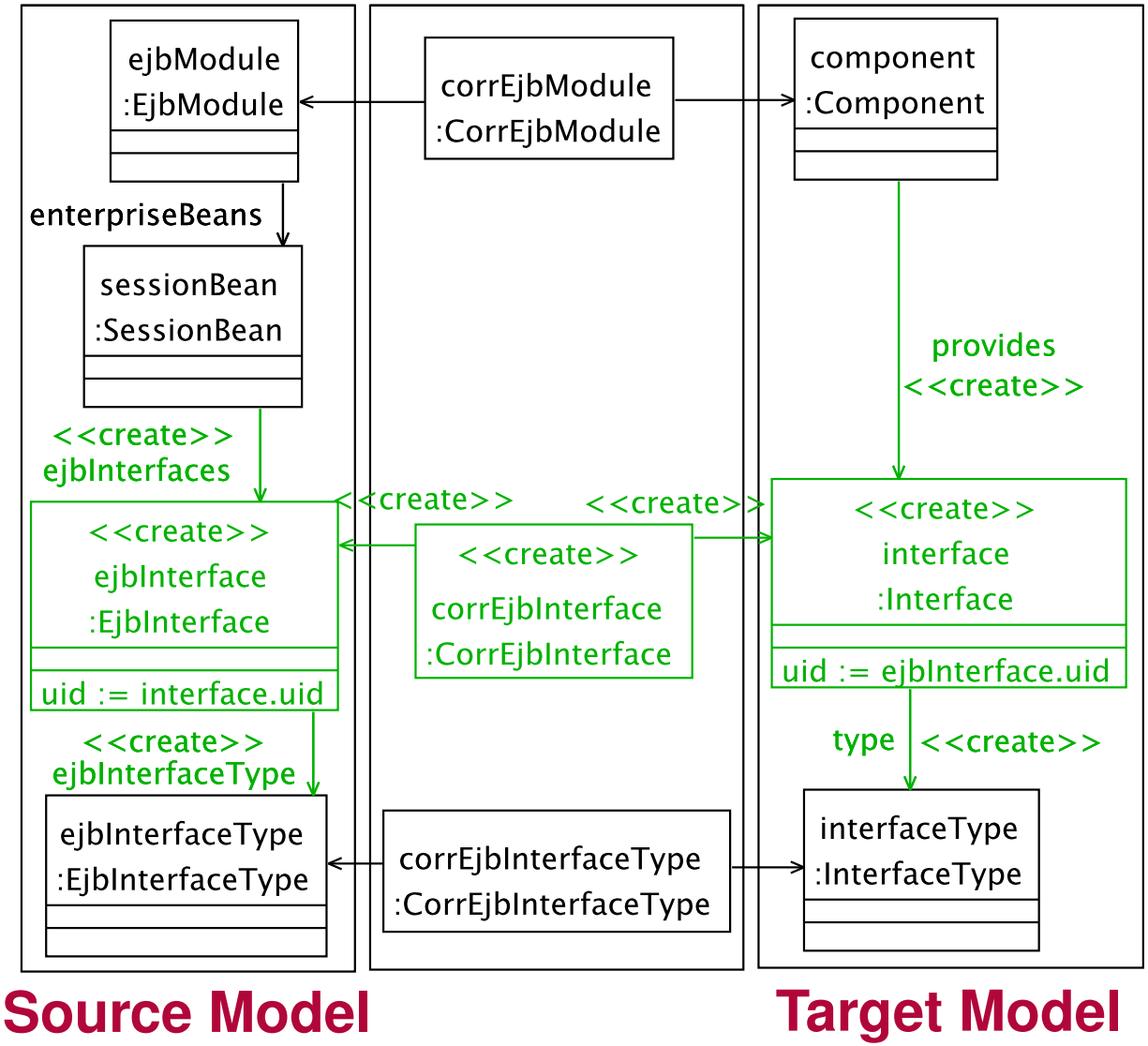
Failure Metamodel
(Target Metamodel)

Model Transformation Engine

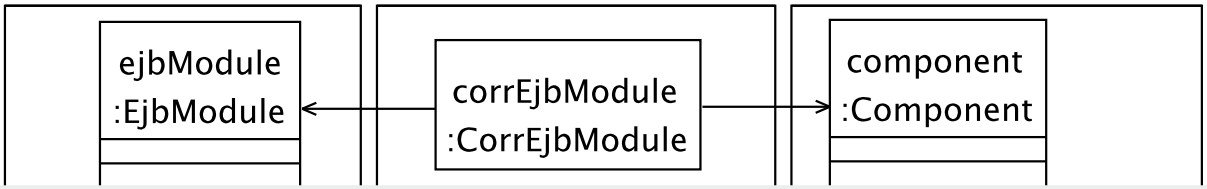


- Unidirectional vs. Bidirectional
- Transformation vs. Synchronization
- Bidirectional synchronization based on **Triple Graph Grammars** (TGG) [Giese and Wagner, 2009, Giese and Hildebrandt, 2008]
- **Incremental** and event-driven solution applicable at runtime

Triple Graph Grammar Rule

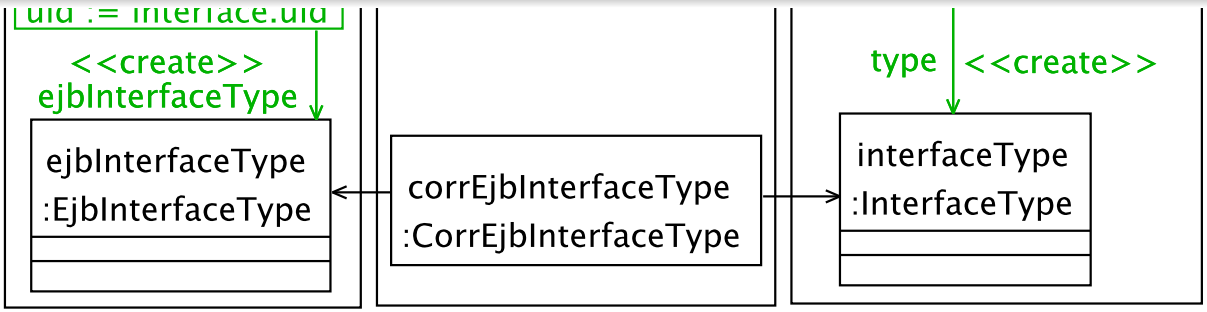


Triple Graph Grammar Rule



- Declarative rules
- Automatic generation of operational rules
- Abstraction gap between models: manually written code “extending” the rules for adaptation

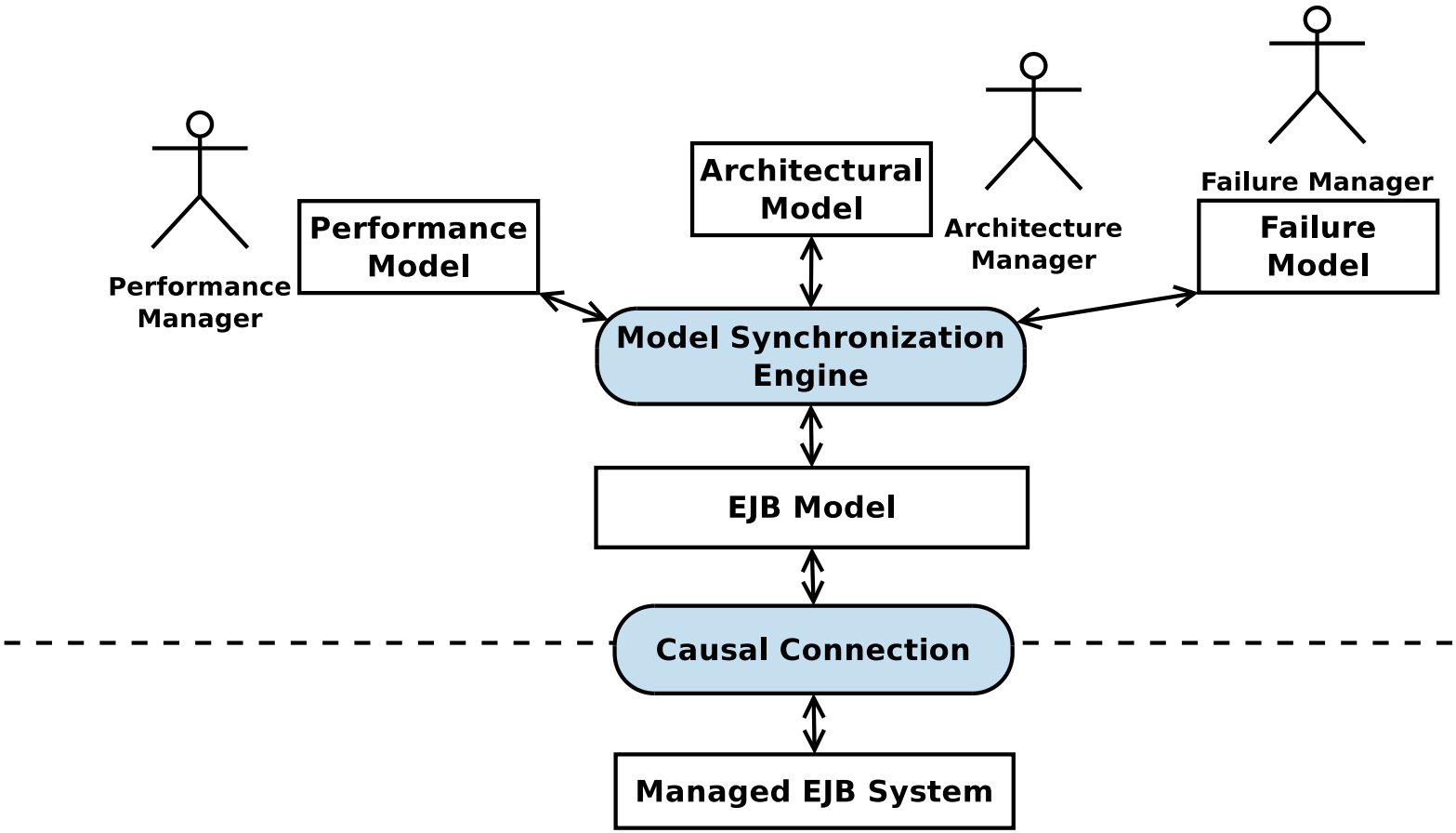
→ **MDE simplifies the development of maintaining multiple runtime models**



Source Model **Target Model**

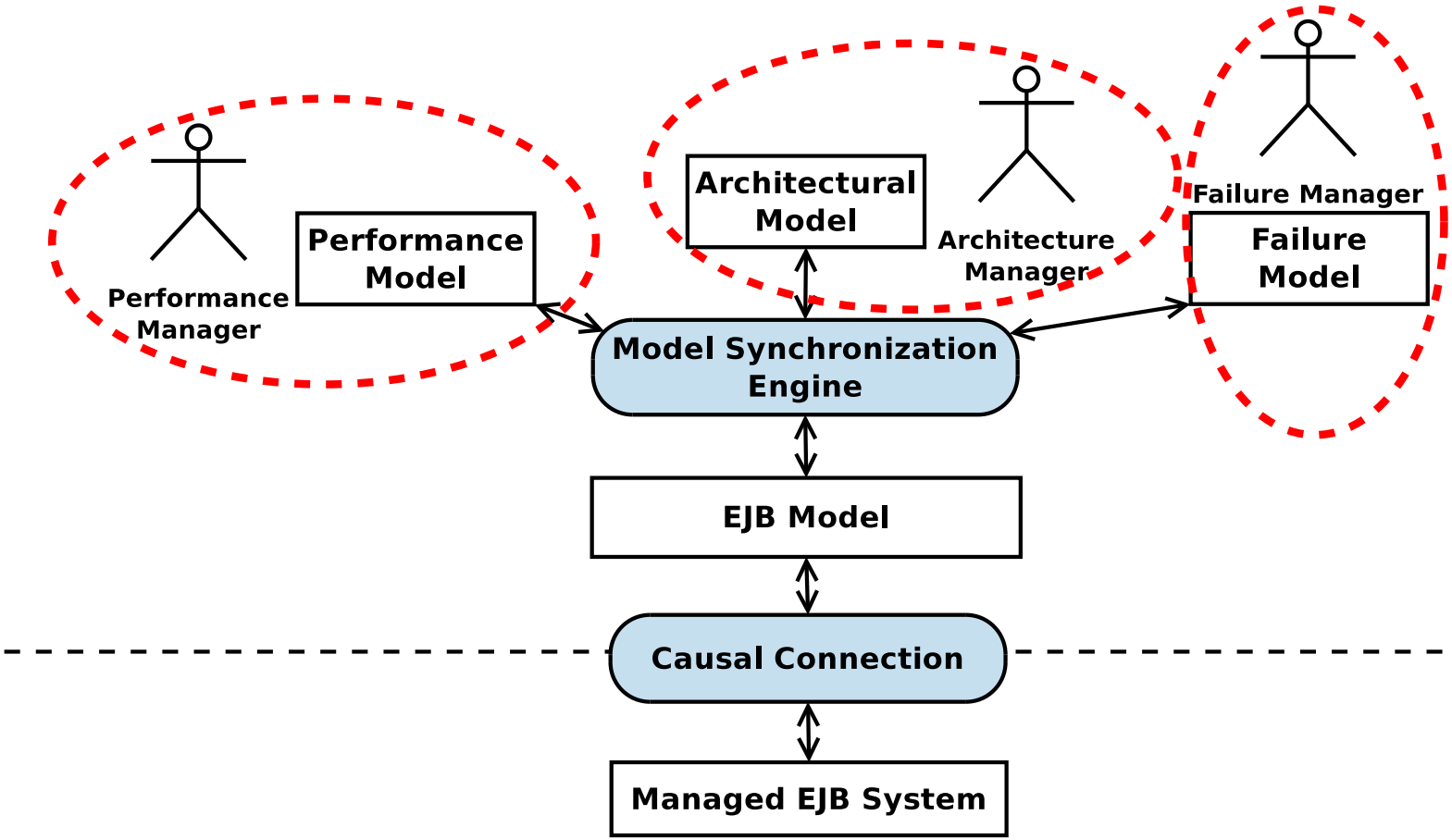
Multiple Runtime Models...

...but how they are **related**?



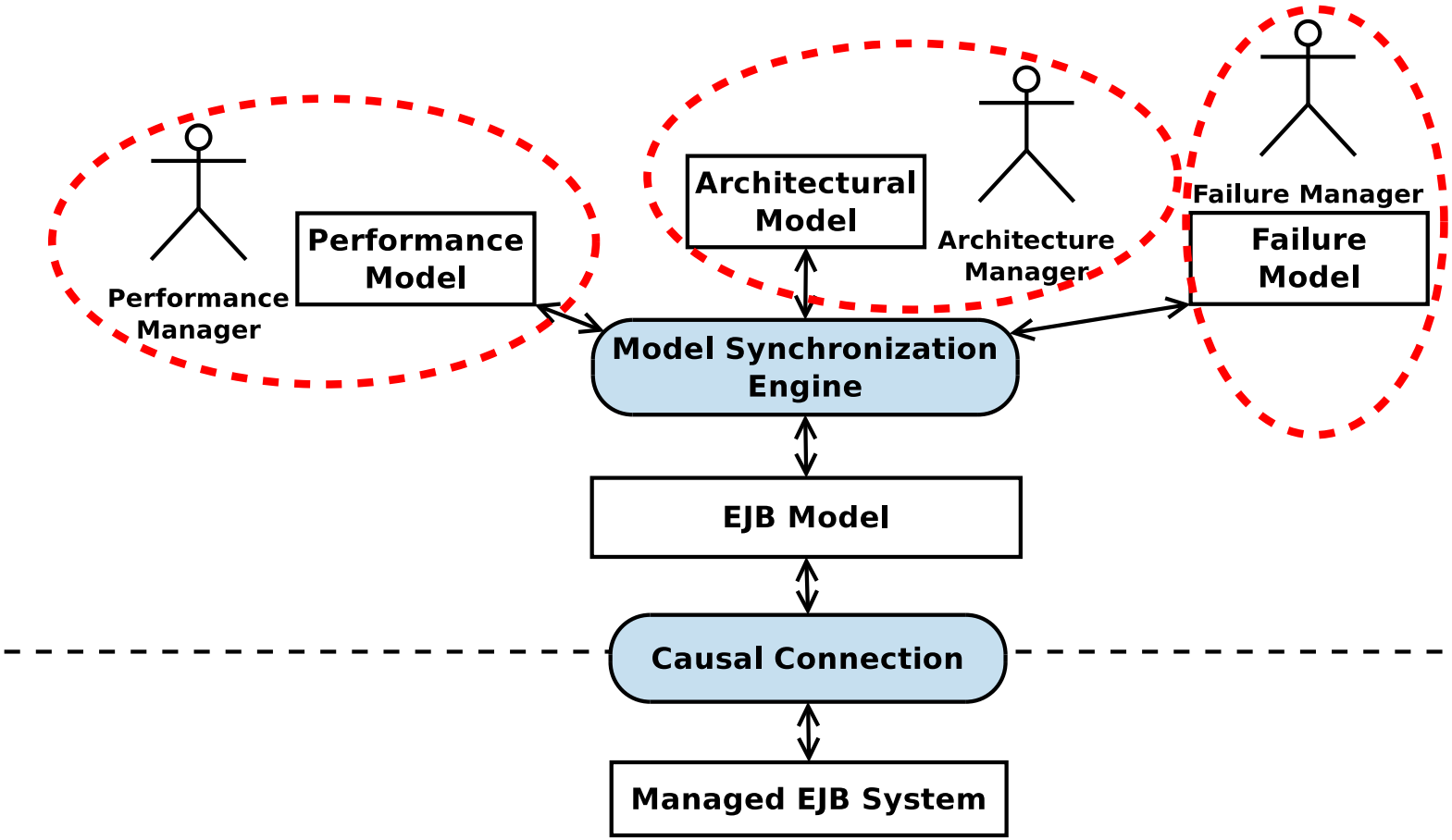
Multiple Runtime Models...

...but how they are **related**? **dependencies, trade-offs,...?**



Multiple Runtime Models…

…but how they are **related**? **dependencies, trade-offs,…?**

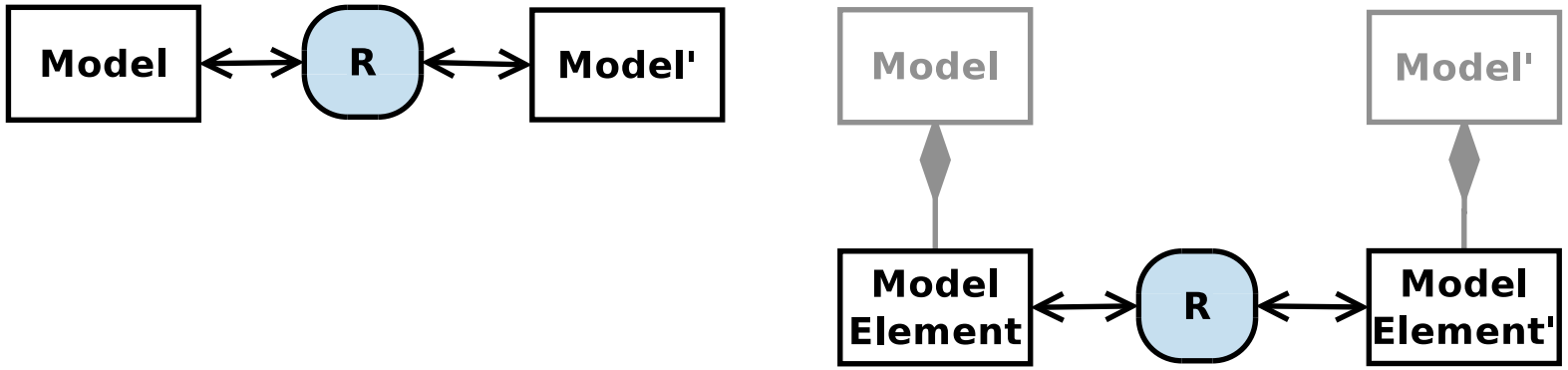


…and how to describe and utilize relations?

Megamodels

"Good enough" Definition (Megamodel)

A *megamodel* is a model that contains models and relations between those models or between elements of those models.



- Makes relations explicit
- Basis for **model-driven** management of models and relations
- Research by Favre [Favre, 2005] and Bézivin et al. [Bézivin et al., 2003, Bézivin et al., 2004, Barbero et al., 2007]

Megamodel Concepts

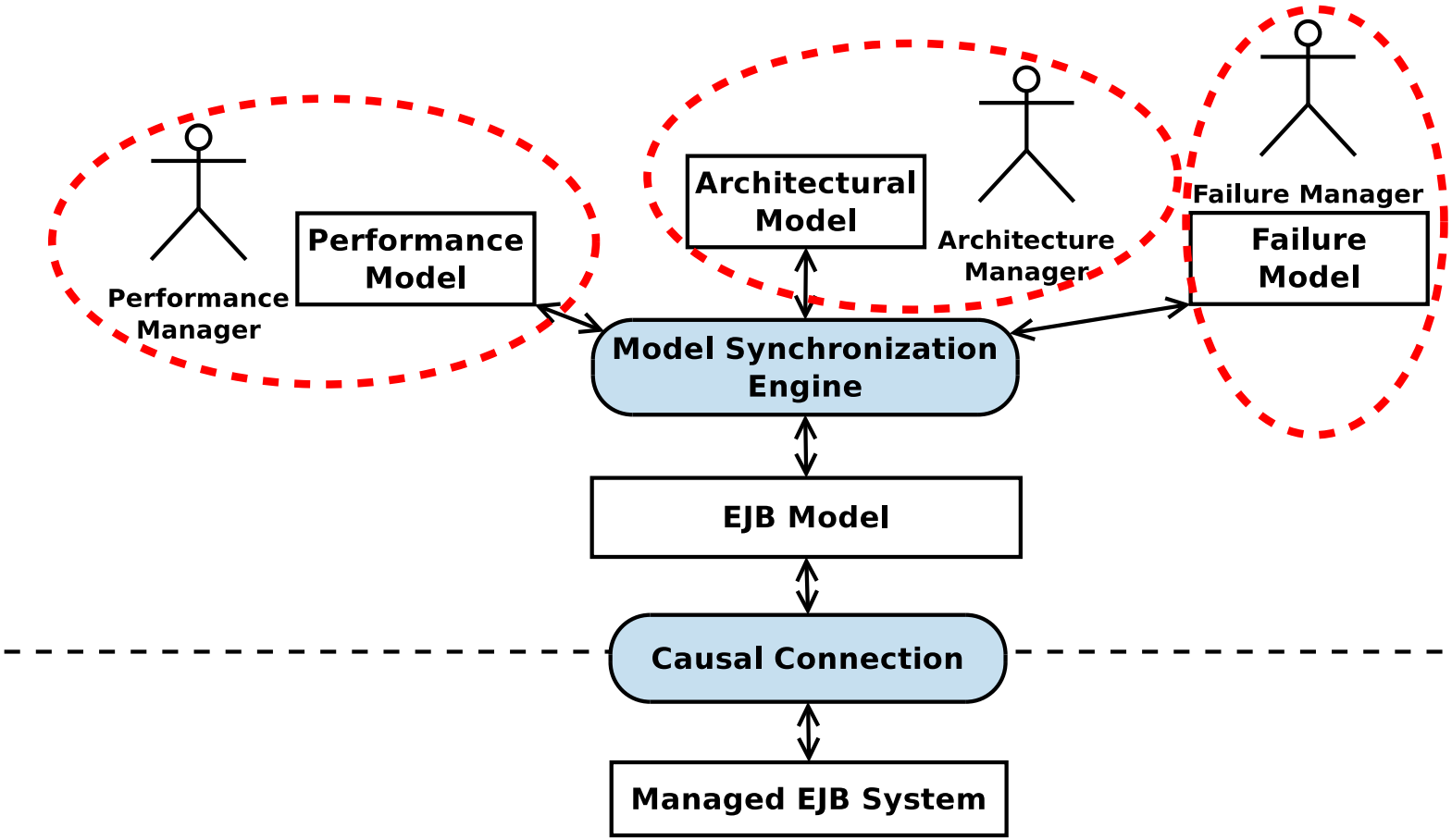
Organizational Purposes:

- Organizing and structuring models and relations
- Registry for models and their relations

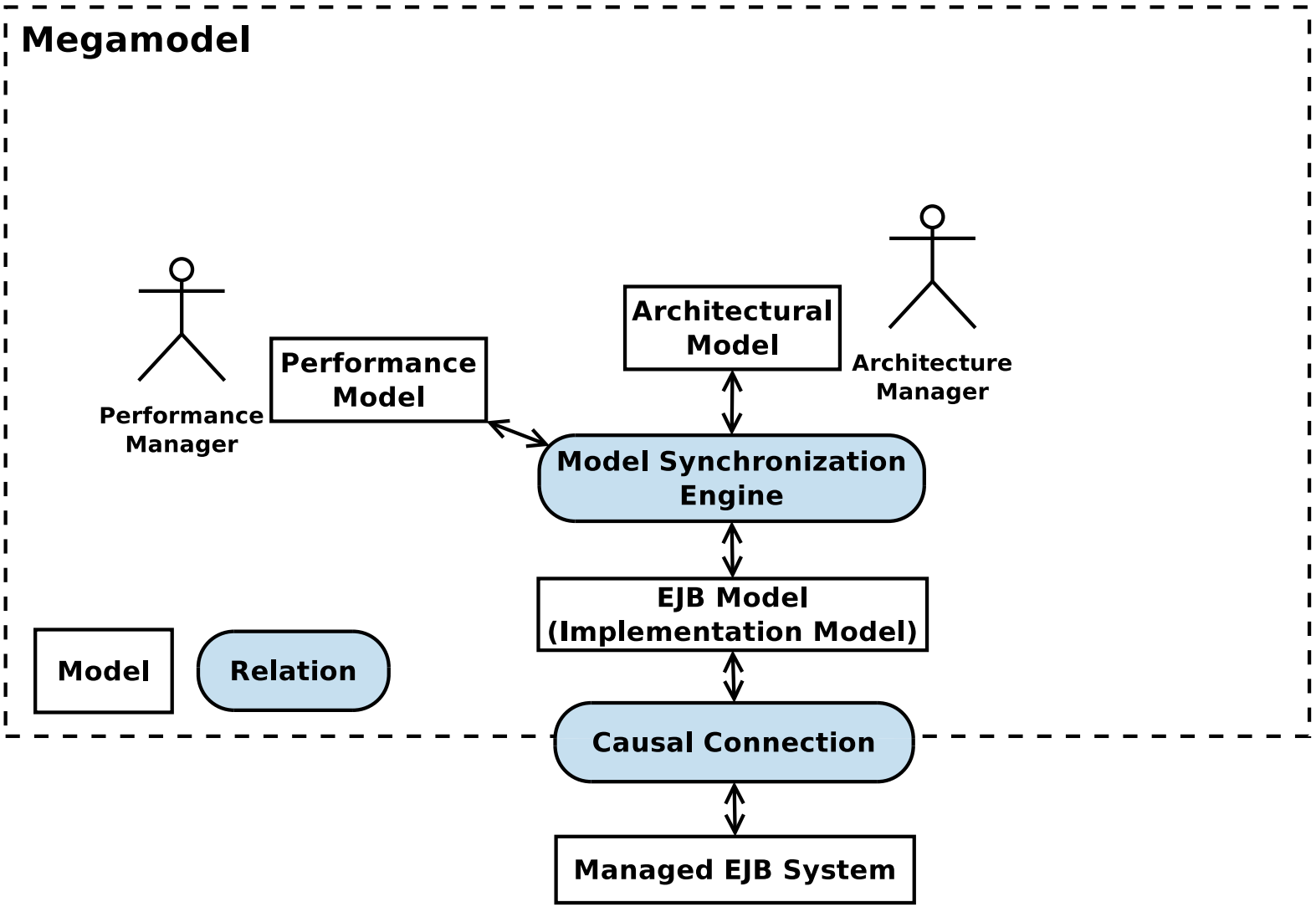
Utilization Purposes:

- Navigation through different models in a model-driven manner
- Operational relations by means of executable units

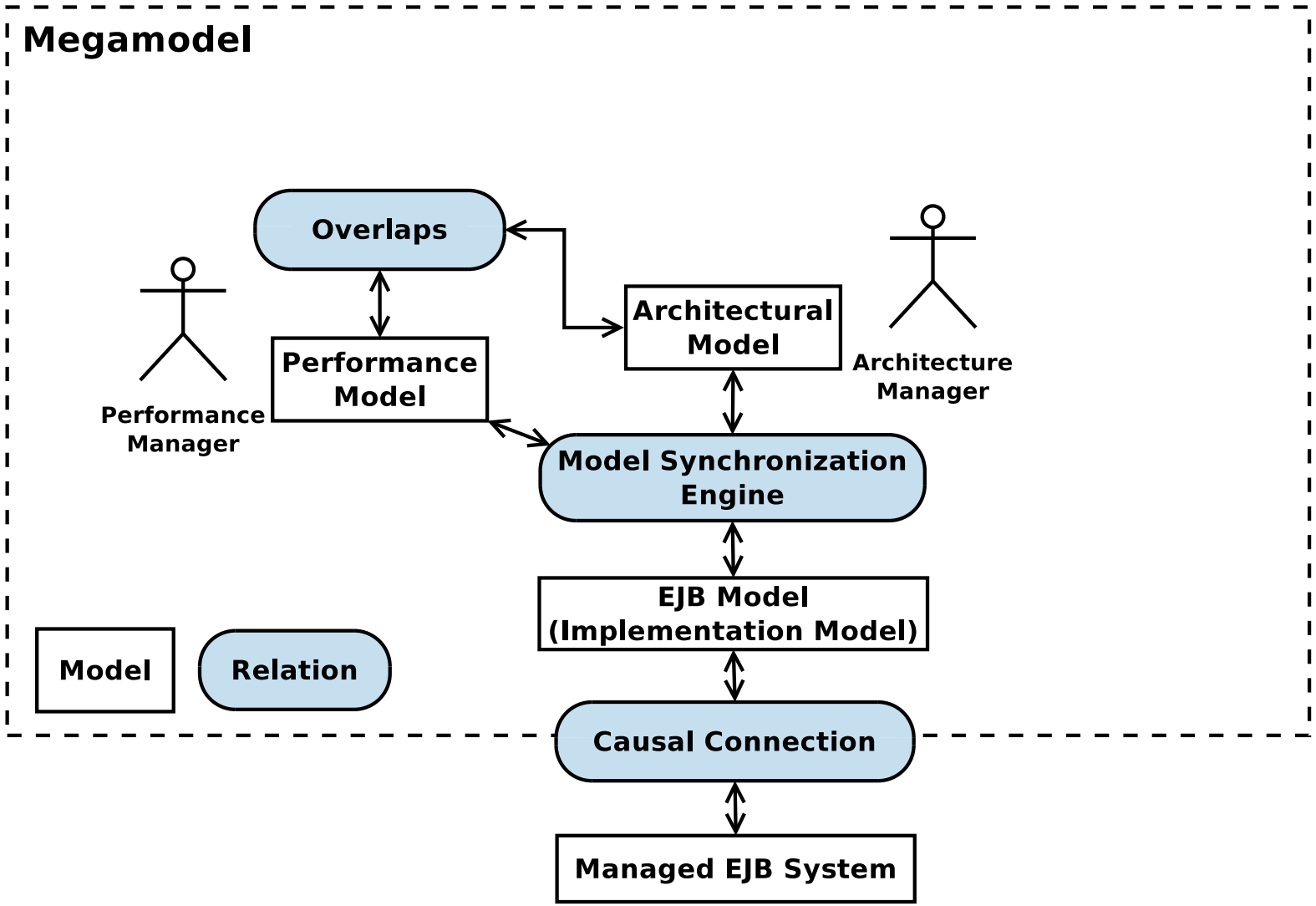
Self-Managing EJB System



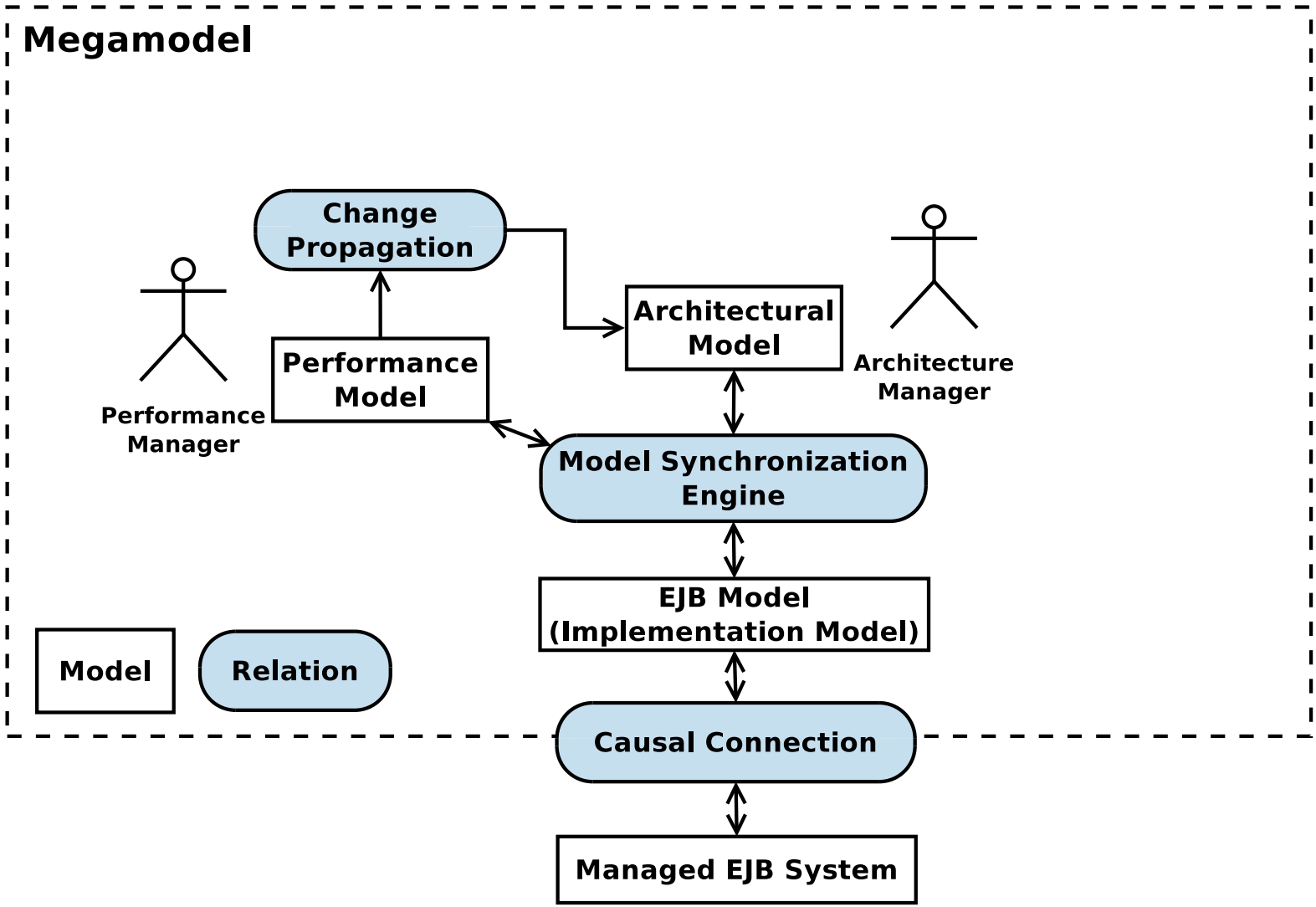
Self-Managing EJB System



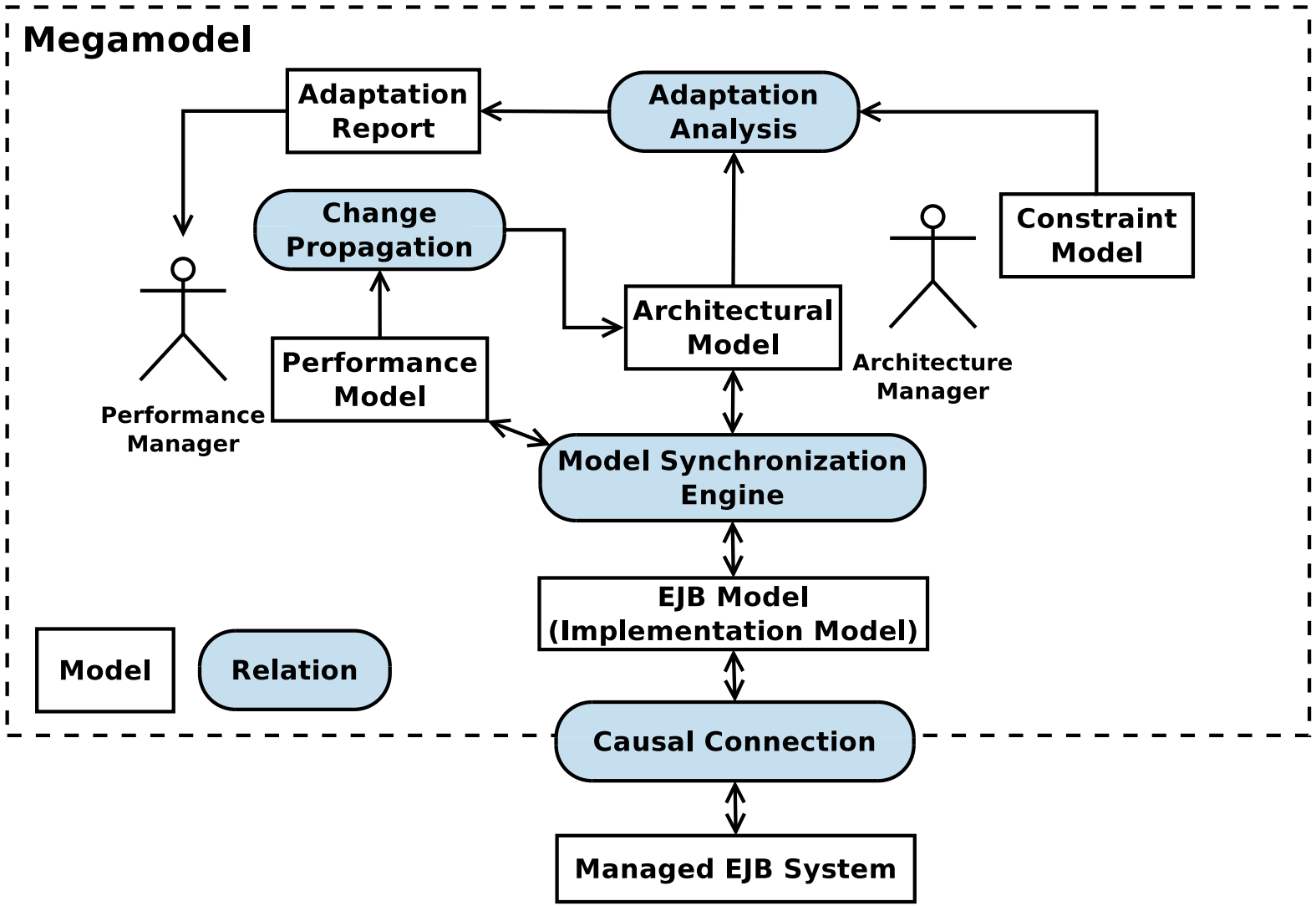
Self-Managing EJB System



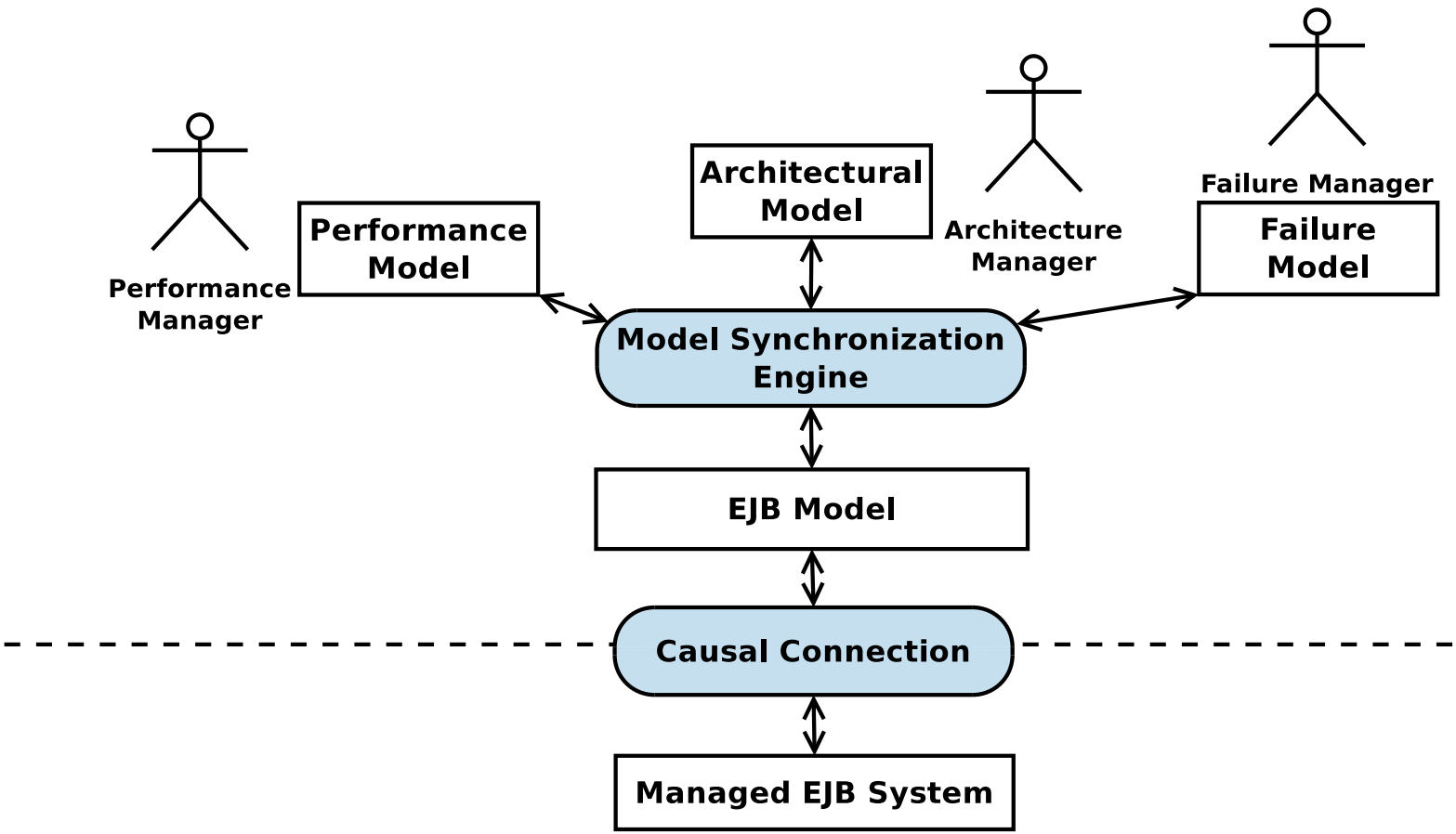
Self-Managing EJB System



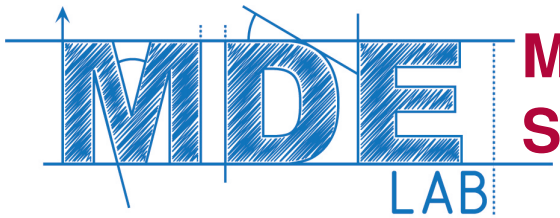
Self-Managing EJB System



Implementation

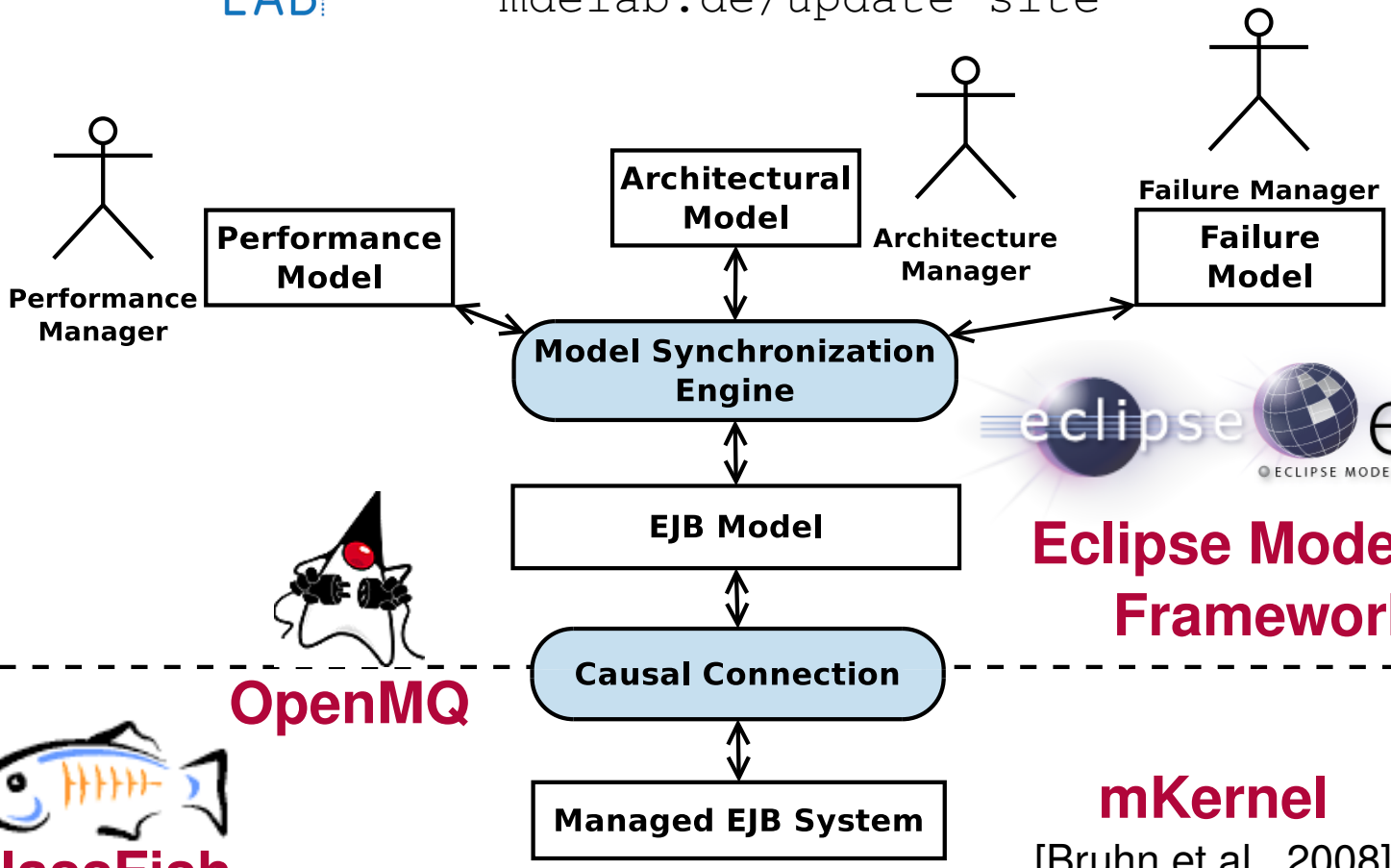


Implementation



Model Transformation Engine (MoTE)
Story Diagram Modeling (SDM) Tools

mdelab.de/update-site



Conclusion and Future Work

Conclusion

- Working approach using **multiple runtime models** for architectural monitoring and adaptation
- **Model Synchronization** techniques to maintain these models
- Initial ideas on megamodel concepts at runtime

Conclusion and Future Work

Conclusion

- Working approach using **multiple runtime models** for architectural monitoring and adaptation
- **Model Synchronization** techniques to maintain these models
- Initial ideas on megamodel concepts at runtime

Future Work

- Architecting self-managing systems
- Semantics of models and model operations
- Describing and utilizing relations at runtime

Conclusion and Future Work

Conclusion

- Working approach using **multiple runtime models** for architectural monitoring and adaptation
- **Model Synchronization** techniques to maintain these models
- Initial ideas on megamodel concepts at runtime

Future Work

- Architecting self-managing systems
- Semantics of models and model operations
- Describing and utilizing relations at runtime

Thank You!

References I

[Barbero et al., 2007] Barbero, M., Fabro, M. D. D., and Bézivin, J. (2007).
Traceability and Provenance Issues in Global Model Management.
In *ECMDA-TW'07: Proc. of 3rd Workshop on Traceability*, pages 47–55.

[Bézivin et al., 2003] Bézivin, J., Gérard, S., Muller, P.-A., and Rioux, L. (2003).
MDA components: Challenges and Opportunities.
In *1st Intl. Workshop on Metamodelling for MDA*, pages 23–41.

[Bézivin et al., 2004] Bézivin, J., Jouault, F., and Valduriez, P. (2004).
On the Need for Megamodels.
In *Proc. of the OOPSLA/GPCE Workshop on Best Practices for Model-Driven Software Development*.

[Blair et al., 2009] Blair, G., Bencomo, N., and France, R. B. (2009).
Models@run.time: Guest Editors' Introduction.
Computer, 42(10):22–27.

[Bruhn et al., 2008] Bruhn, J., Niklaus, C., Vogel, T., and Wirtz, G. (2008).
Comprehensive support for management of Enterprise Applications.
In *Proceedings of the 6th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2008), Doha, Katar*, pages 755–762. IEEE Computer Society.

[Cheng et al., 2009] Cheng, B. H., de Lemos, R., Giese, H., Inverardi, P., Magee, J., and et al. (2009).
Software Engineering for Self-Adaptive Systems: A Research Road Map.
In *Software Engineering for Self-Adaptive Systems*, volume 5525 of *LNCS*, pages 1–26. Springer.

[Diao et al., 2005] Diao, Y., Hellerstein, J. L., Parekh, S., Griffith, R., Kaiser, G., and Phung, D. (2005).
Self-Managing Systems: A Control Theory Foundation.
In *ECBS '05: Proceedings of the 12th IEEE International Conference and Workshops on Engineering of Computer-Based Systems*, pages 441–448, Washington, DC, USA. IEEE Computer Society.

[Favre, 2005] Favre, J.-M. (2005).
Foundations of Model (Driven) (Reverse) Engineering : Models – Episode I: Stories of The Fidus Papyrus and of The Solarus.
In *Language Engineering for Model-Driven Software Development*, number 04101 in Dagstuhl Seminar Proceedings. IBFI, Schloss Dagstuhl.

[France and Rumpe, 2007] France, R. and Rumpe, B. (2007).
Model-driven Development of Complex Software: A Research Roadmap.
In *Proc. of the ICSE Workshop on Future of Software Engineering (FOSE)*, pages 37–54. IEEE.

[Giese and Hildebrandt, 2008] Giese, H. and Hildebrandt, S. (2008).
Incremental Model Synchronization for Multiple Updates.
In *Proc. of the 3rd Intl. Workshop on Graph and Model Transformation*. ACM.

[Giese and Wagner, 2009] Giese, H. and Wagner, R. (2009).
From Model Transformation to Incremental Bidirectional Model Synchronization.
Software and Systems Modeling, 8(1).

[Kephart and Chess, 2003] Kephart, J. and Chess, D. (2003).
The Vision of Autonomic Computing.
IEEE Computer, 36(1):41–50.

[Kokar et al., 1999] Kokar, M. M., Baclawski, K., and Eracar, Y. A. (1999).
Control Theory-Based Foundations of Self-Controlling Software.
Intelligent Systems and their Applications, IEEE, 14(3):37–45.

References II

[Lehman, 1996] Lehman, M. M. (1996).
Laws of Software Evolution Revisited.
In Montangero, C., editor, *Software Process Technology, 5th European Workshop, EWSPT'96, Nancy, France, October 9-11, 1996, Proceedings*, volume 1149 of *Lecture Notes in Computer Science*, pages 108–124. Springer.

[Lin et al., 2005] Lin, P., MacArthur, A., and Leaney, J. (2005).
Defining Autonomic Computing: A Software Engineering Perspective.
In *ASWEC '05: Proceedings of the 2005 Australian conference on Software Engineering*, pages 88–97, Washington, DC, USA. IEEE Computer Society.

[Northrop et al., 2006] Northrop, L., Feiler, P. H., Gabriel, R. P., Linger, R., Longstaff, T., Kazman, R., Klein, M., and Schmidt, D. (2006).
Ultra-Large-Scale Systems: The Software Challenge of the Future.
Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

[Sommerville, 2007] Sommerville, I. (2007).
Software Engineering.
Addison Wesley, 8 edition.

[Sterritt, 2005] Sterritt, R. (2005).
Autonomic computing.
Innovations in Systems and Software Engineering, 1(1):79–88.

[Vogel and Giese, 2010] Vogel, T. and Giese, H. (2010).
Adaptation and Abstract Runtime Models.
In *Proceedings of the 5th Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2010) at the 32nd IEEE/ACM International Conference on Software Engineering (ICSE 2010), Cape Town, South Africa*, pages 39–48. ACM.

[Vogel et al., 2009a] Vogel, T., Neumann, S., Hildebrandt, S., Giese, H., and Becker, B. (2009a).
Incremental Model Synchronization for Efficient Run-time Monitoring.
In Bencomo, N., Blair, G., France, R., Jeanneret, C., and Munoz, F., editors, *Proceedings of the 4th International Workshop on Models@run.time at the 12th IEEE/ACM International Conference on Model Driven Engineering Languages and Systems (MoDELS 2009), Denver, Colorado, USA*, volume 509 of *CEUR Workshop Proceedings*, pages 1–10. CEUR-WS.org.
(best paper).

[Vogel et al., 2009b] Vogel, T., Neumann, S., Hildebrandt, S., Giese, H., and Becker, B. (2009b).
Model-Driven Architectural Monitoring and Adaptation for Autonomic Systems.
In *Proceedings of the 6th IEEE/ACM International Conference on Autonomic Computing and Communications (ICAC 2009), Barcelona, Spain*, pages 67–68. ACM.

[Vogel et al., 2010a] Vogel, T., Neumann, S., Hildebrandt, S., Giese, H., and Becker, B. (2010a).
Incremental Model Synchronization for Efficient Run-Time Monitoring.
In Ghosh, S., editor, *Models in Software Engineering, Workshops and Symposia at MODELS 2009, Denver, CO, USA, October 4-9, 2009, Reports and Revised Selected Papers*, volume 6002 of *Lecture Notes in Computer Science (LNCS)*, pages 124–139. Springer-Verlag.

[Vogel et al., 2010b] Vogel, T., Seibel, A., and Giese, H. (2010b).
Toward Megamodels at Runtime.
In Bencomo, N., Blair, G., Fleurey, F., and Jeanneret, C., editors, *Proceedings of the 5th International Workshop on Models@run.time at the 13th IEEE/ACM International Conference on Model Driven Engineering Languages and Systems (MoDELS 2010), Oslo, Norway*, volume 641 of *CEUR Workshop Proceedings*, pages 13–24. CEUR-WS.org.
(best paper).