



UNIVERSITATEA
DIN BUCUREȘTI

Metode de dezvoltare software

Arhitecturi software

08.04.2025

Arhitecturi software

Arhitectura unui sistem

- Arhitectura software este un subdomeniu important al ingineriei software
- Aceasta reprezintă **împărțirea optimă a unui sistem complex în diverse componente, evidențiind relațiile dintre acestea.**
- este esențială pentru a avea un sistem funcțional și scalabil
- de știut: jobul de "arhitect software" este bine văzut și bine plătit: https://www.glassdoor.com/List/Highest-Paying-Jobs-LST_KQ0,19.htm

Diverse attribute de calitate

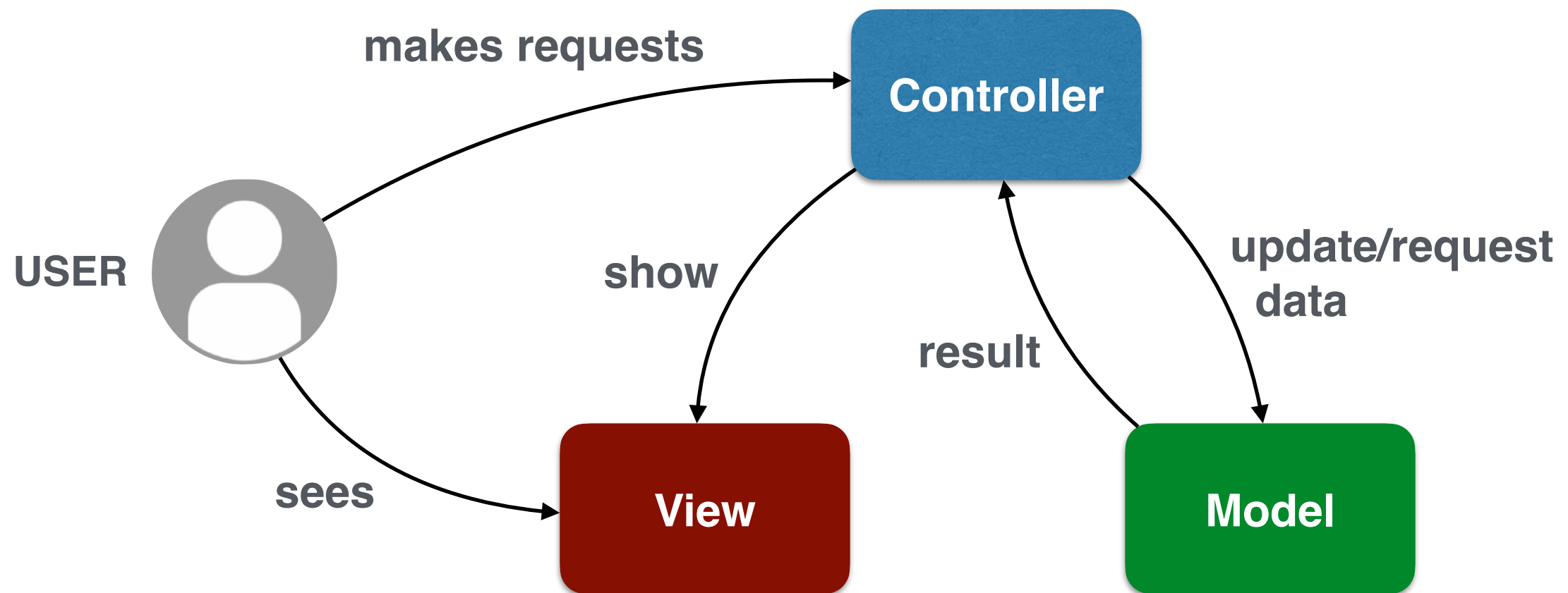
Arhitectura unui sistem trebuie adaptată la attributele de calitate cerute. De exemplu:

- **performanță**: trebuie paralelizat cât mai mult, descompunând sistemul în procese cooperante; trebuie ținute sub control comunicarea și accesul la date
- **precizie**: trebuie optimizată structura datelor și modul în care valorile sunt prelucrate
- **securitate**: trebuie gestionate bine restricțiile de comunicare și acces; trebuie analizate componentele cele mai vulnerabile
- **portabilitate și reutilizare**: trebuie minimizate depedențele puternice între componente.

Stiluri și perspective arhitecturale

- Arhitectura software de obicei prezintă anumite **perspective** ale sistemului, de obicei părțile cele mai importante
- Exemple de perspective:
 - proces
 - data flow
 - comportament
 - deployment
 - relații între module
 - etc.

Arhitectură clasică - Model-View-Controller

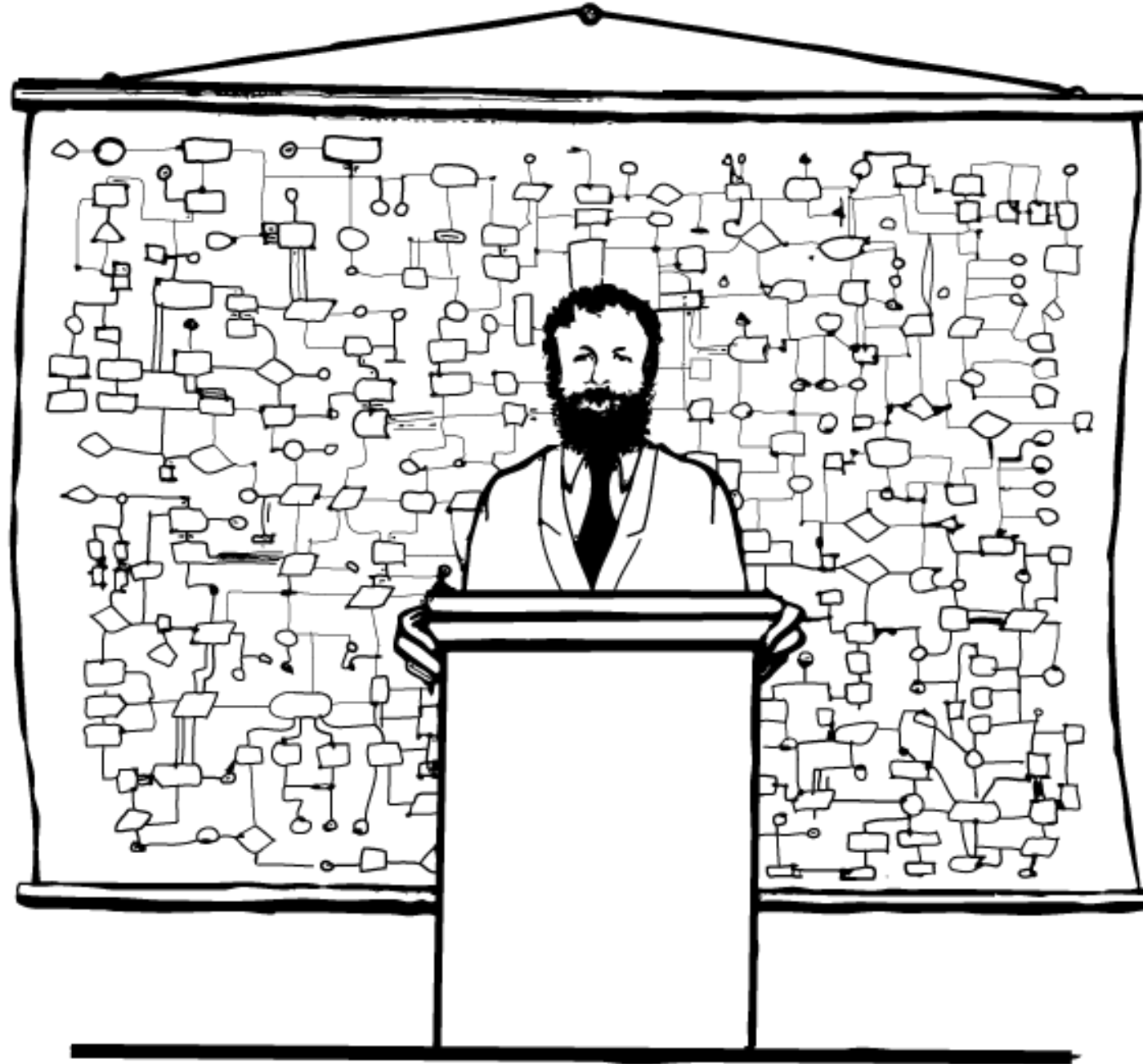


Java Spring MVC: <https://spring.io/guides/gs/serving-web-content>

ASP.NET MVC: <https://docs.microsoft.com/en-us/aspnet/core/mvc>

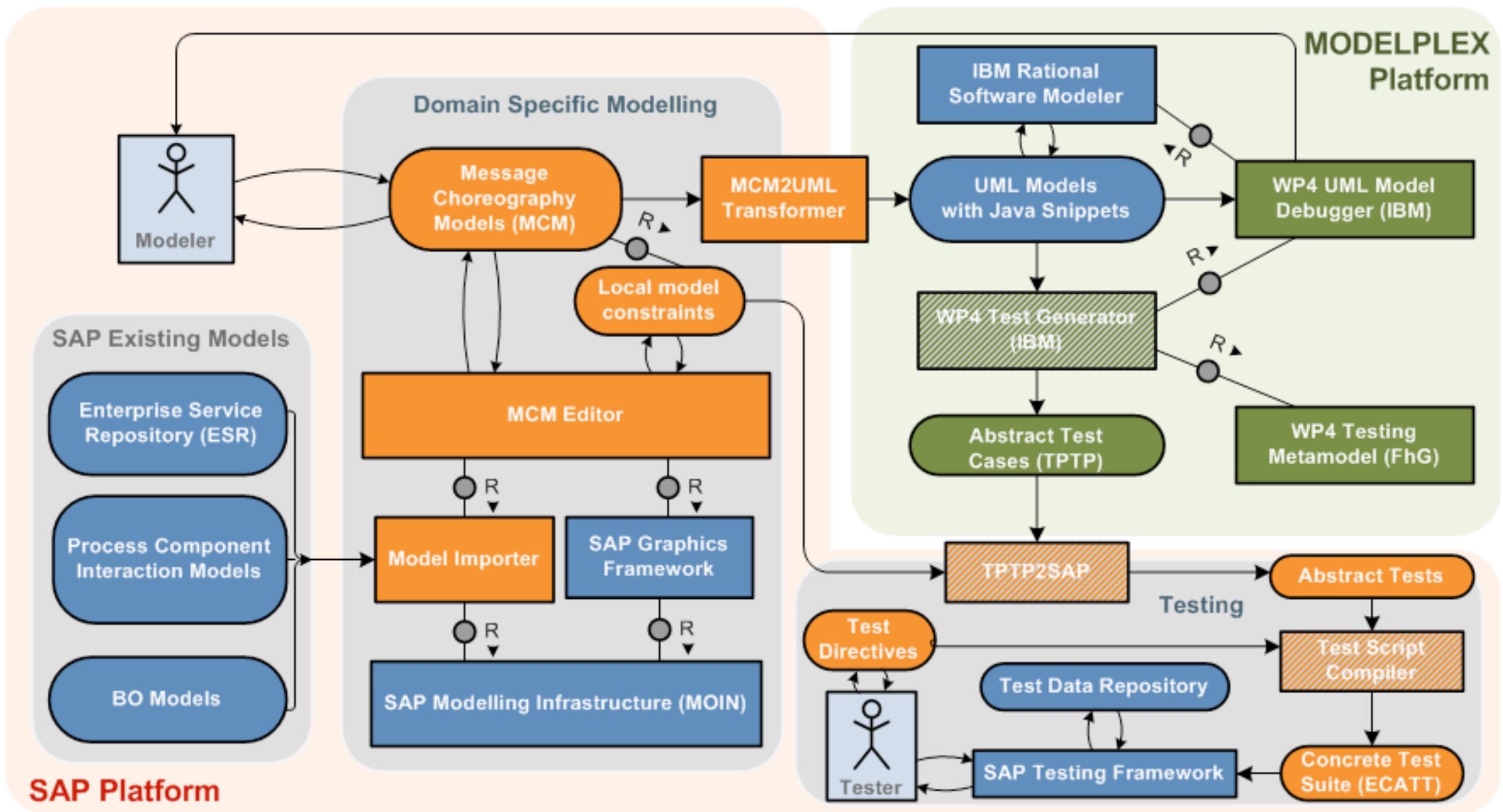
Example....

- în continuare, prezint câteva exemple de arhitecturi cu care am lucrat eu, iar apoi și alte exemple

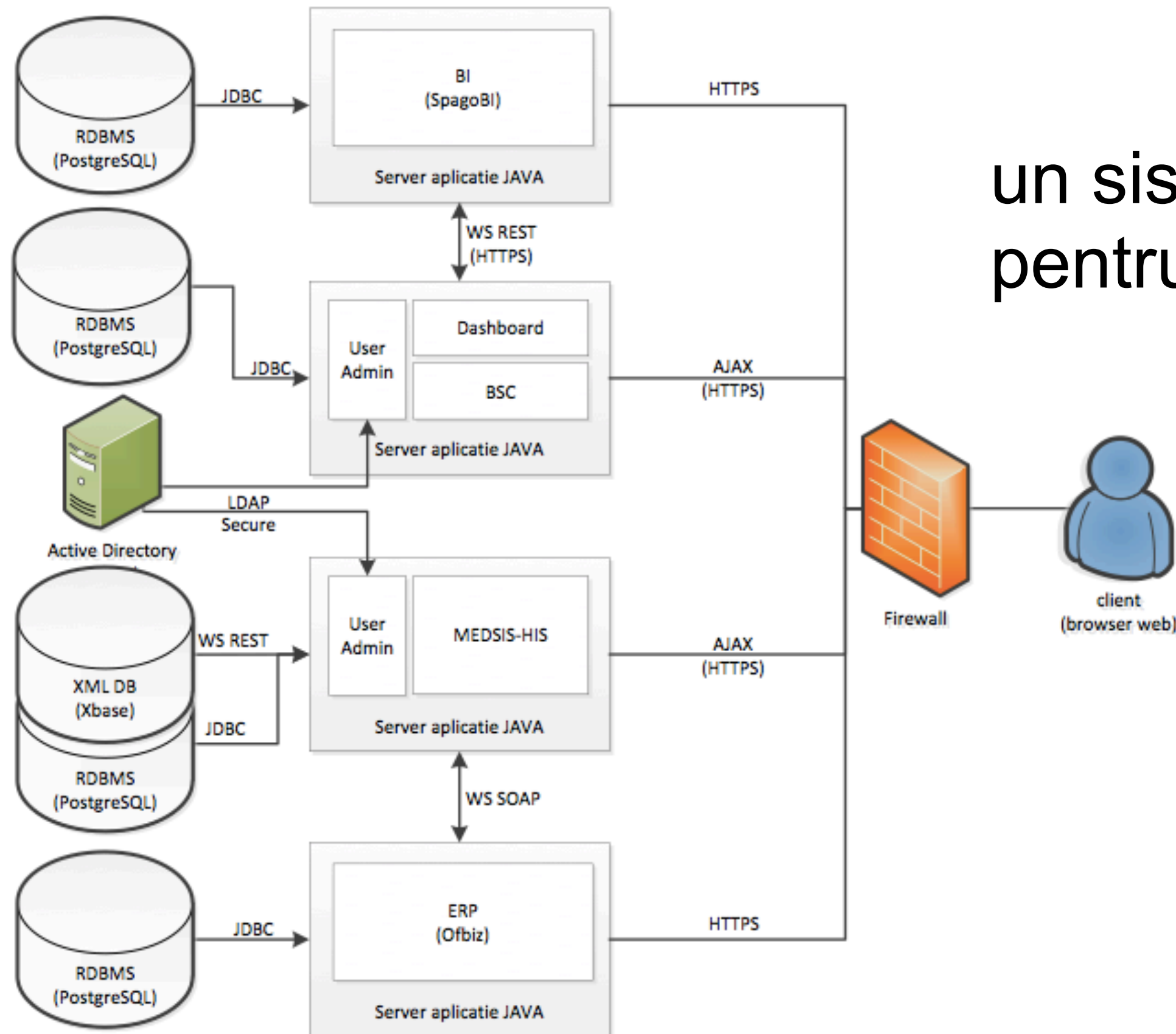


“Now that you have an overview of the system,
we’re ready for a little more detail”

Exemplul 1 - din domeniul enterprise

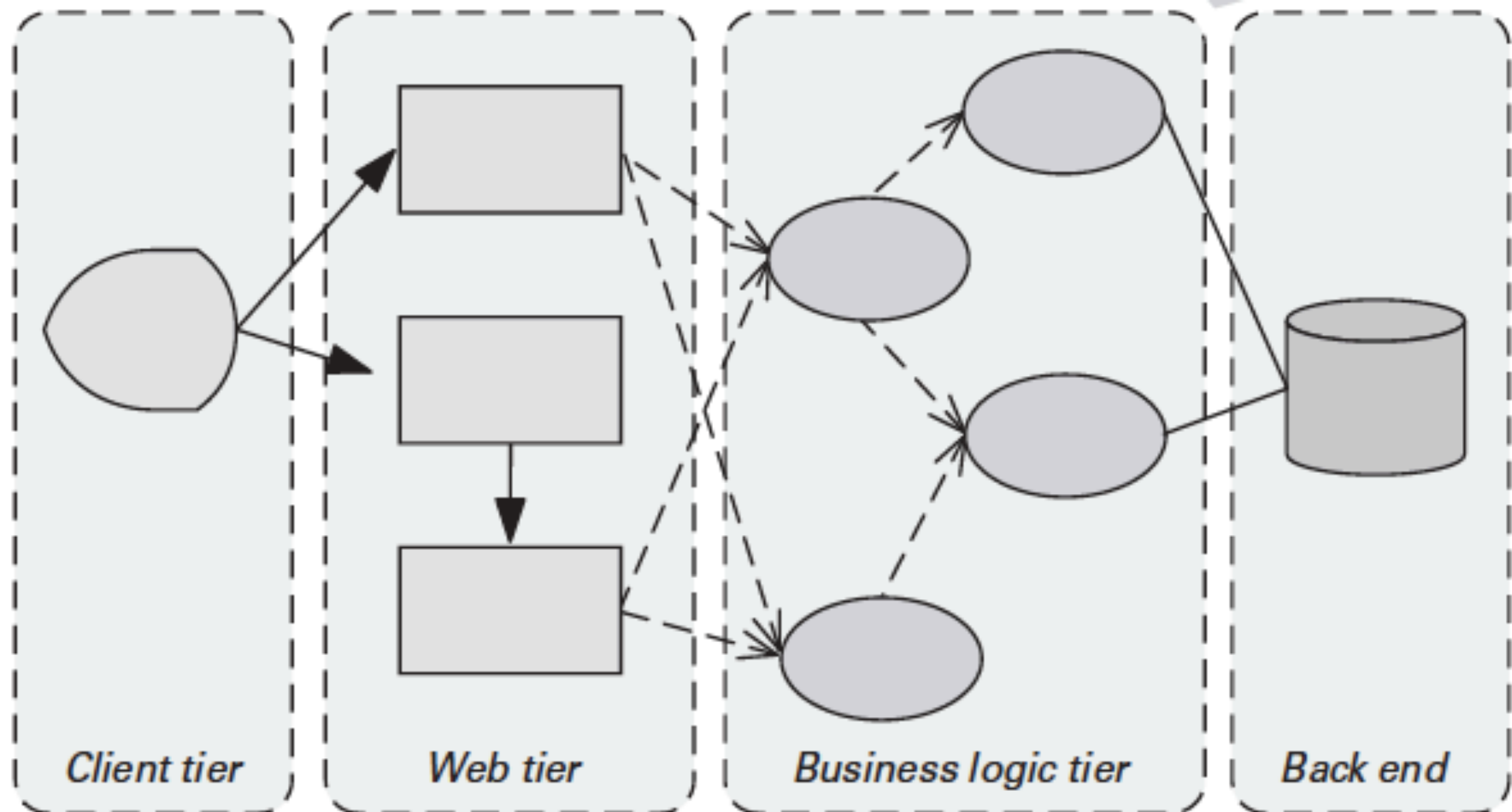


Exemplul 2 - din domeniul medical



un sistem IT
pentru spitale

Exemplul 3 - șablon pt. o aplicație web



Key



Client component



Web component



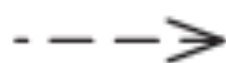
Business component



Database



http/https



Method call



Database access

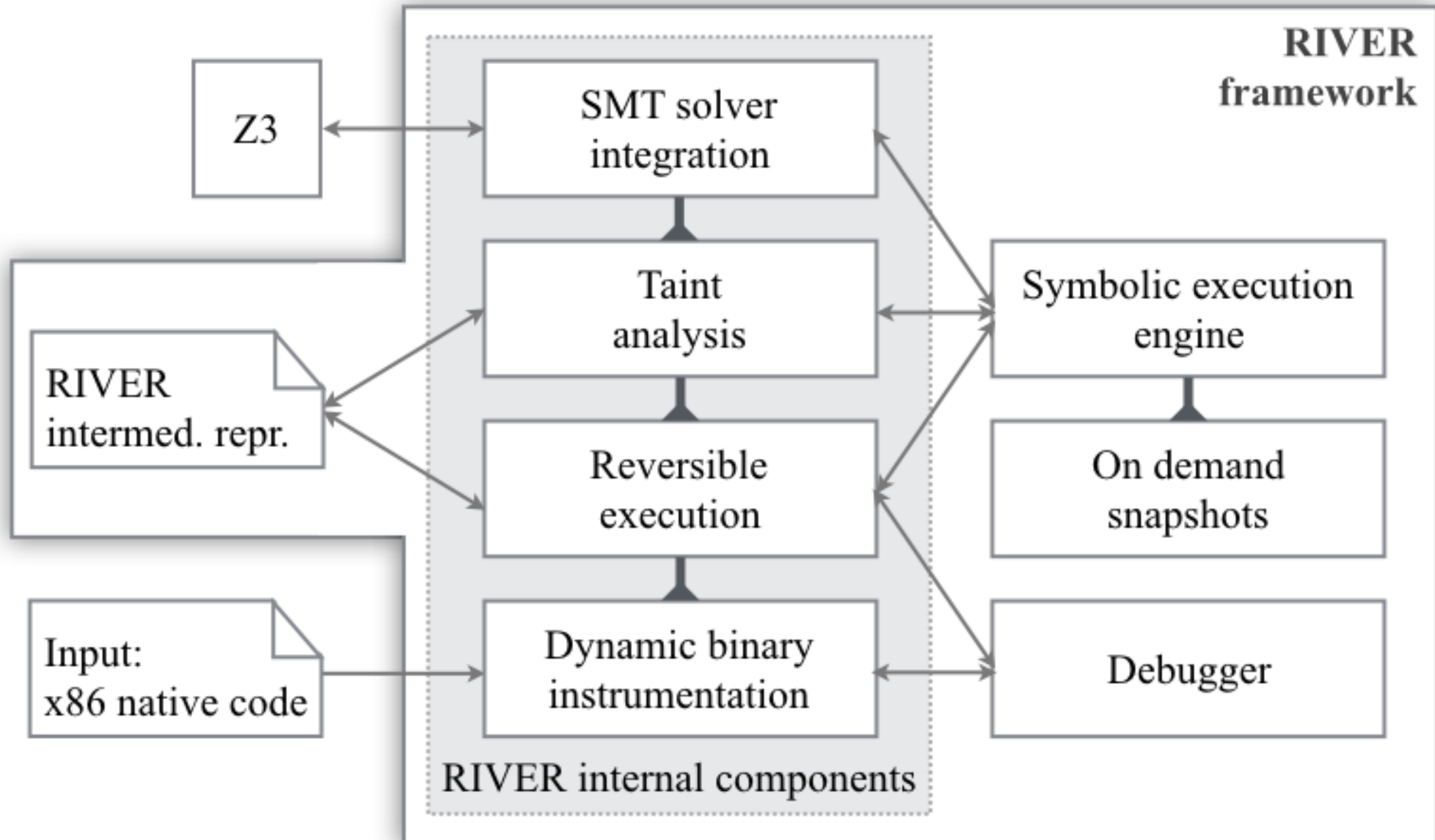


Tier

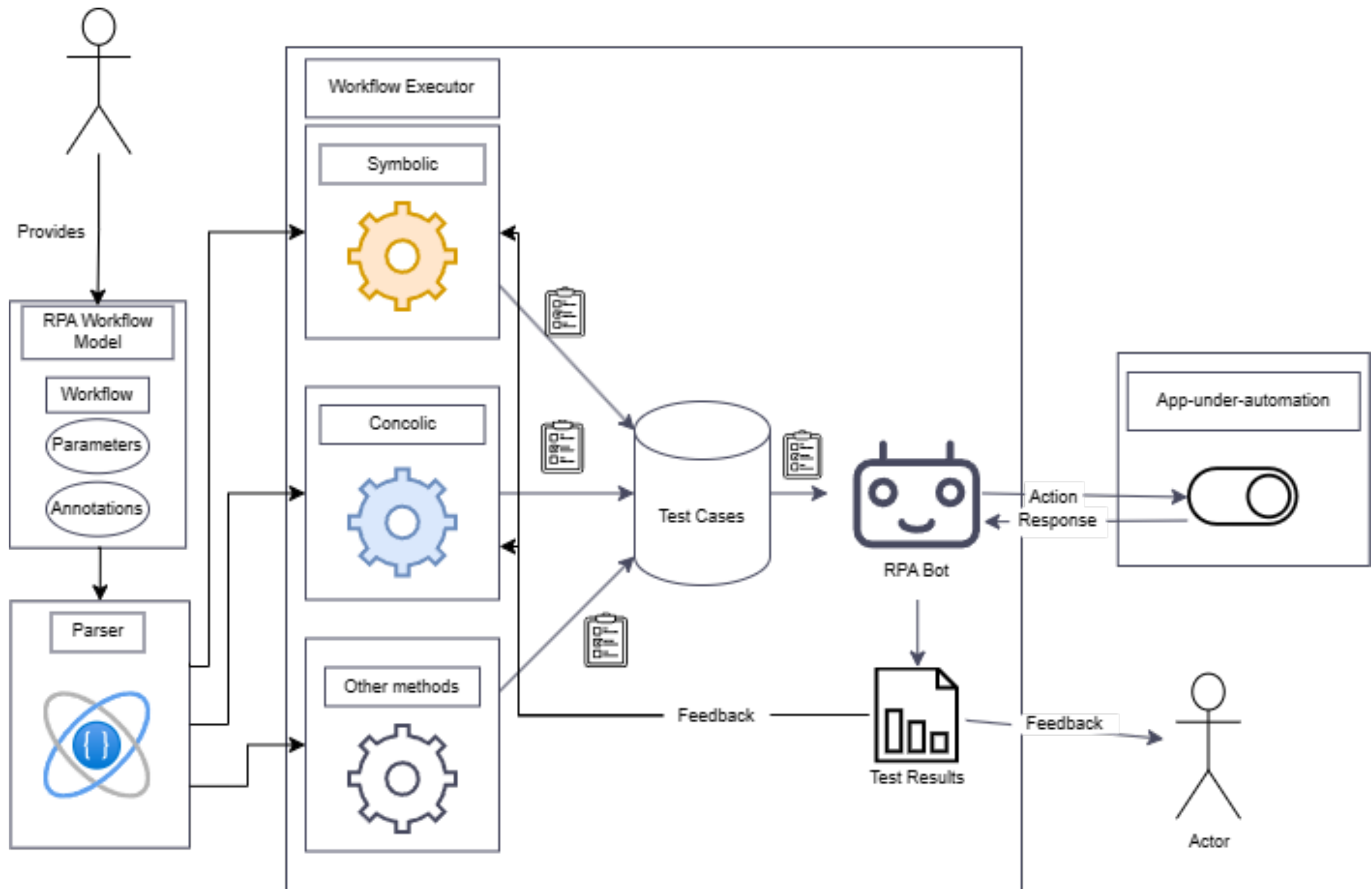
din cartea "Documenting Software Architectures", 2nd ed, 2011 Pearson Publ.

Exemplul 4 - analiza de executabile

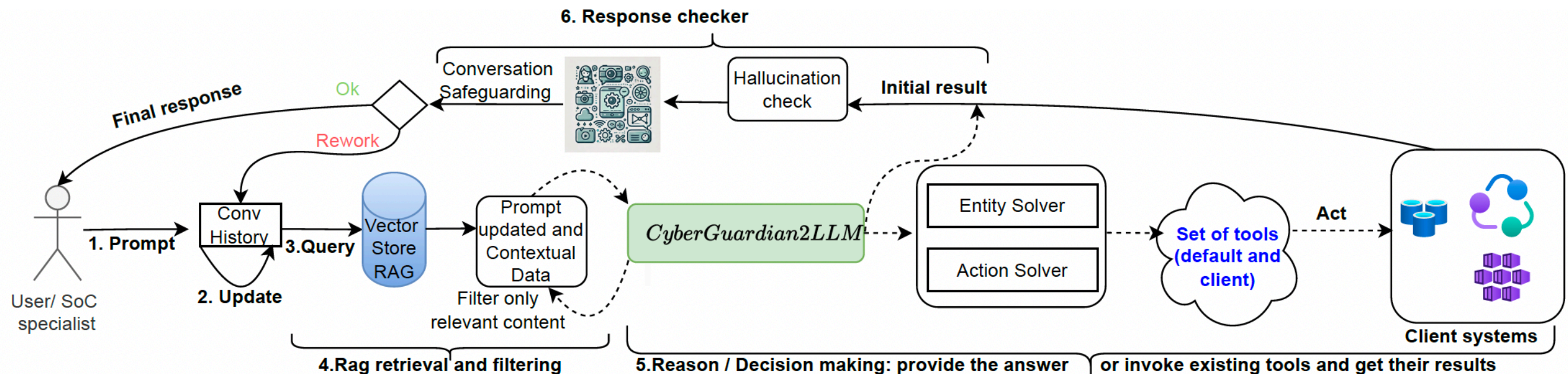
arhitectura unui sistem de analiză dinamică pentru fișiere x86
(dintr-un proiect de cercetare în colaborare Bitdefender - UniBuc)



Exemplu 5 - Tool de testare automată



Exemplu 6 - LLM prompt processing



(1) receiving the user's prompt, (2) reformulating the current query based on the memories of previous conversations, (3) giving the reformulated prompt, retrieving and filtering relevant content from the indexed data, (4) deciding whether the answer can be given or must be given via the exposed services on the client side (using tools/function calls); the decision is based on LLM and some NLP-based processing techniques for easier identification of the entities and actions requested in the prompt, (5) checking the response for hallucinations and security; if the response matches the guidelines, returning it to the user and updating the internal knowledge; if it needs to be revised instead, add the reviewer's response to the history so that the model can assess what was wrong with the original response.

Exemplu 7 - modelare UML a unui joc 2D

https://web.archive.org/web/20150911051854/http://rtb-team.sourceforge.net/rtb-team_analysis.htm

- La linkul de mai sus gasiti un exemplu de proiect software care este documentat prin diverse tipuri de diagrame UML

