

# The $z^2$ Environment - positioned -

# Intro

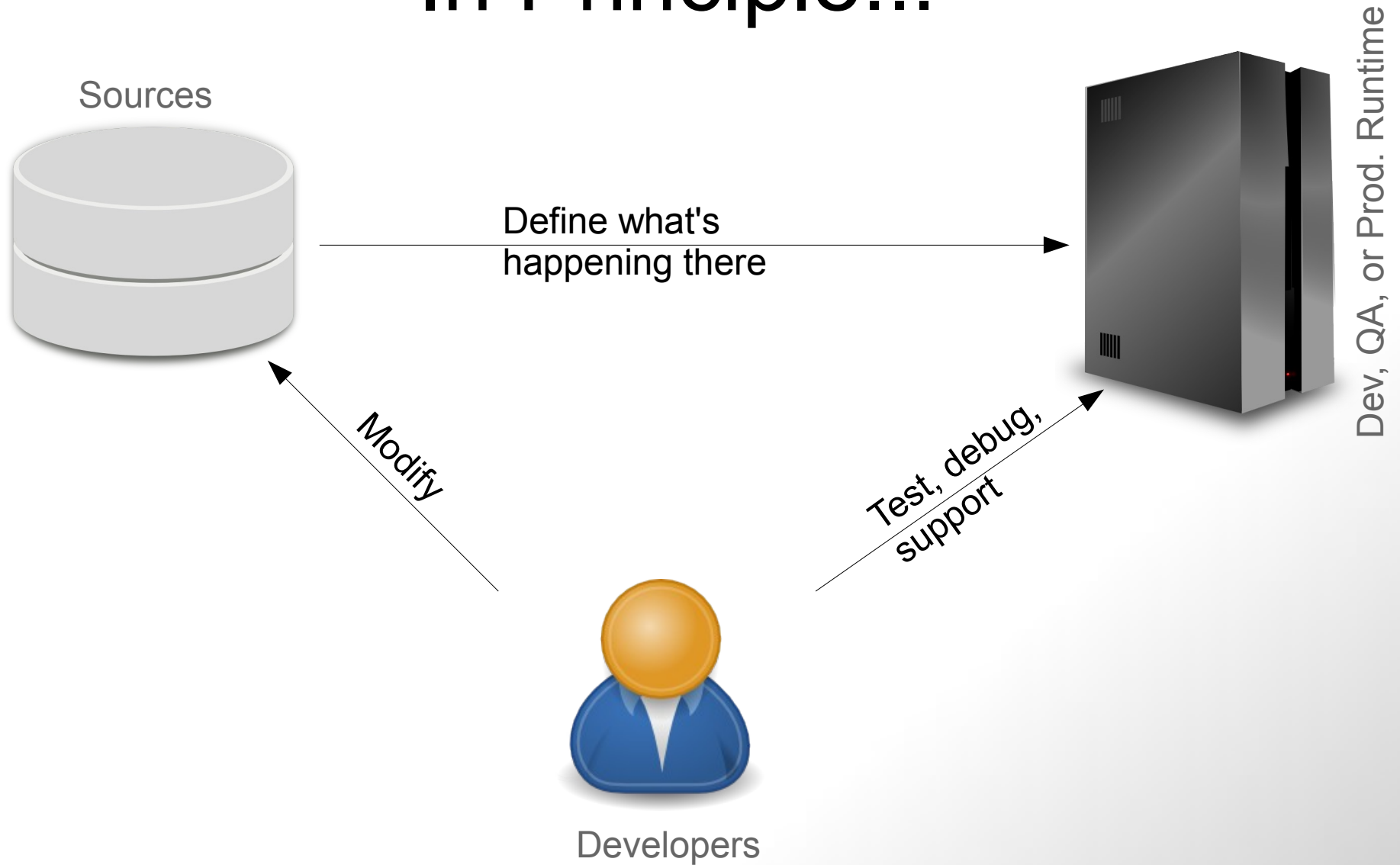
- The z<sup>2</sup> Environment lives on sources
- It removes the complexities of build & deploy
- It implements immediate integration
- We will compare this with
  - A)The classical way of Java team development
  - B)The Maven approach

# What's the Problem?

Have a *consistently up-to-date* execution environment for development and production

*Efficiently* develop and support your solution

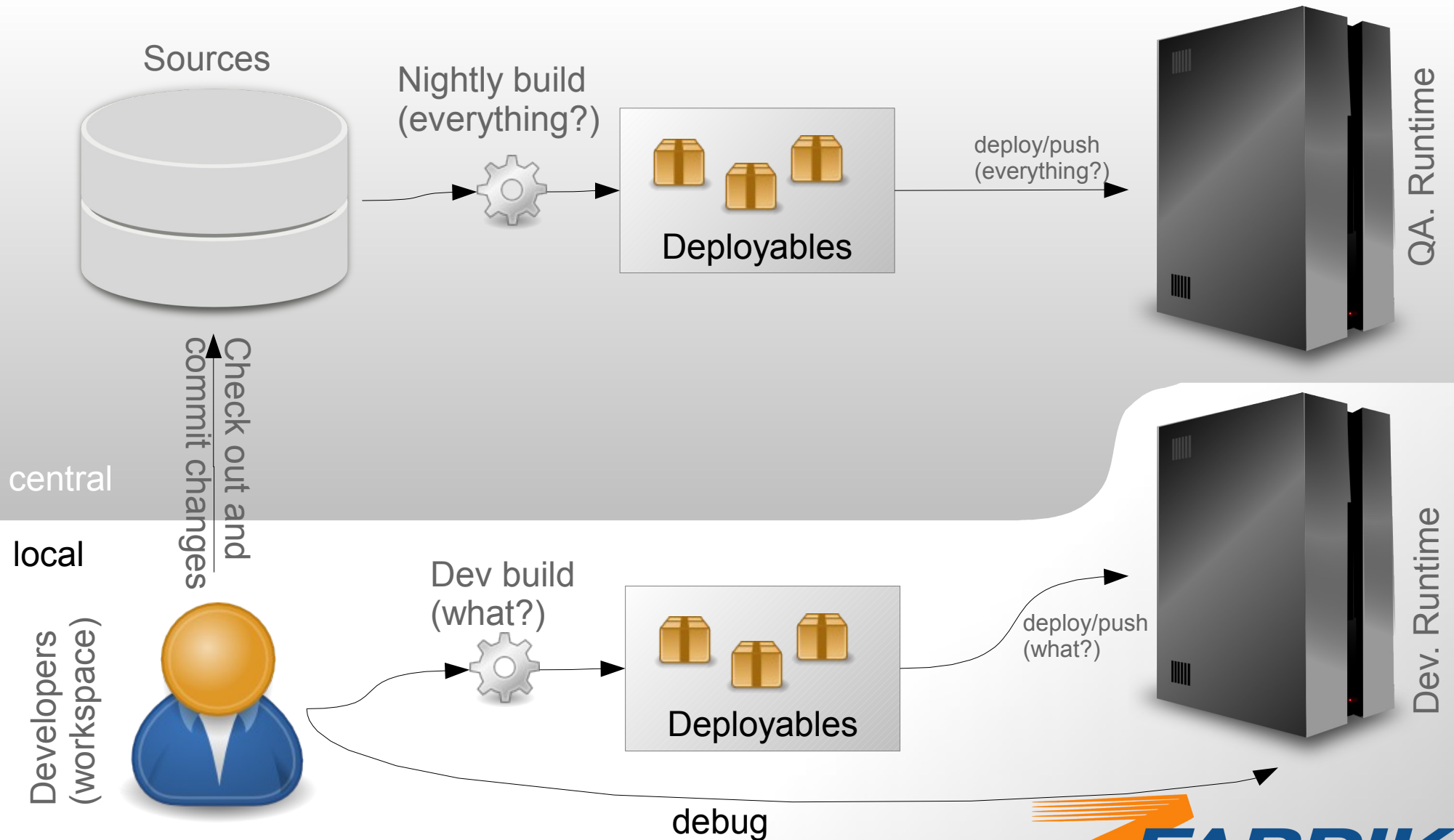
# In Principle...



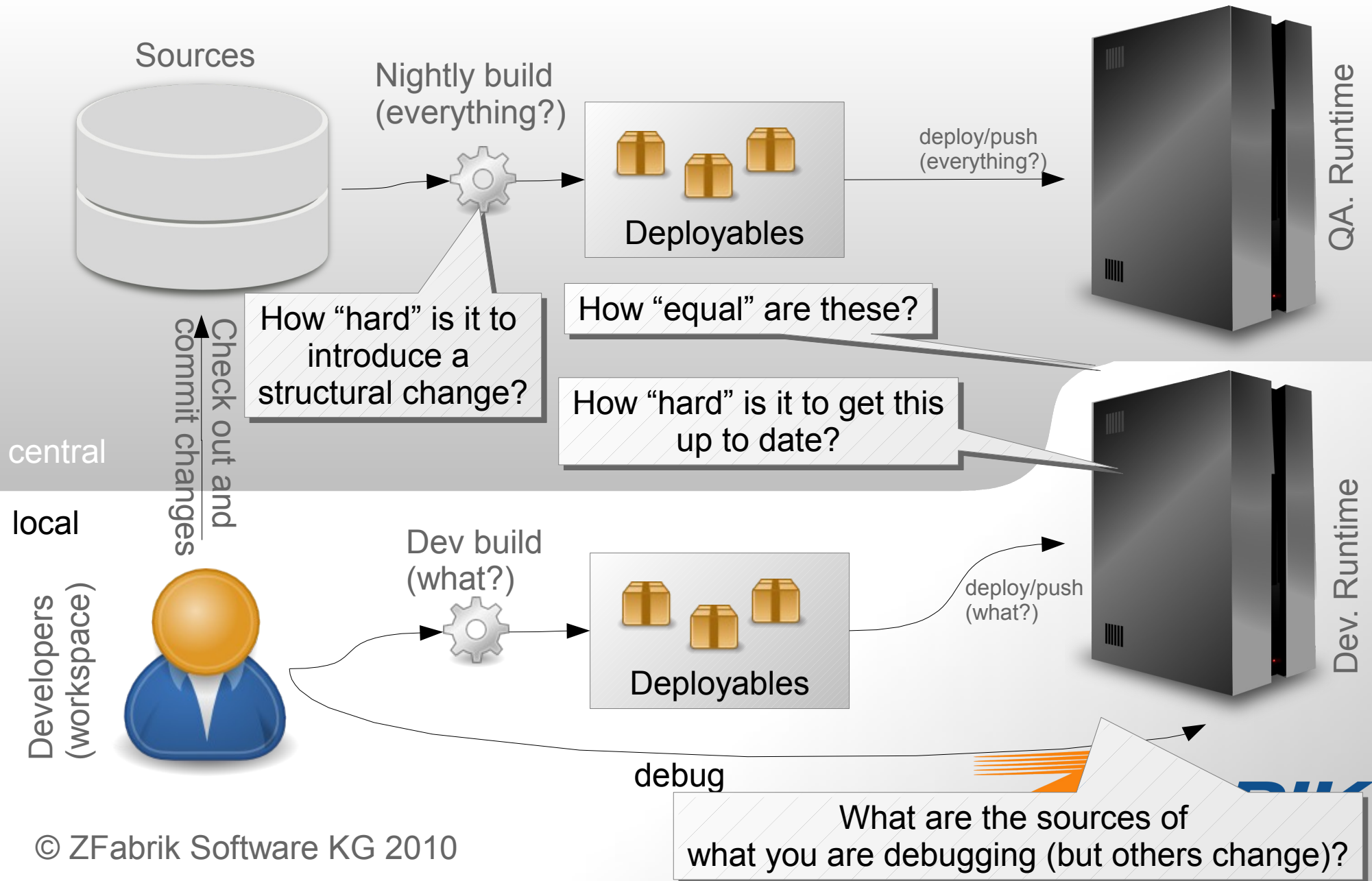
# So, Where is the Problem?

- Java application servers do not live on sources but rather on binaries
- It is up to the development organization to (somehow) provide those binaries and hence...
  - ... find out what binaries are affected by a change
  - ... find out what binaries belong to one solution
  - ... keep development environments up-to-date
  - ... integrate changes (continuously)
  - ... live with delays between commit and run

# The "classical" approach



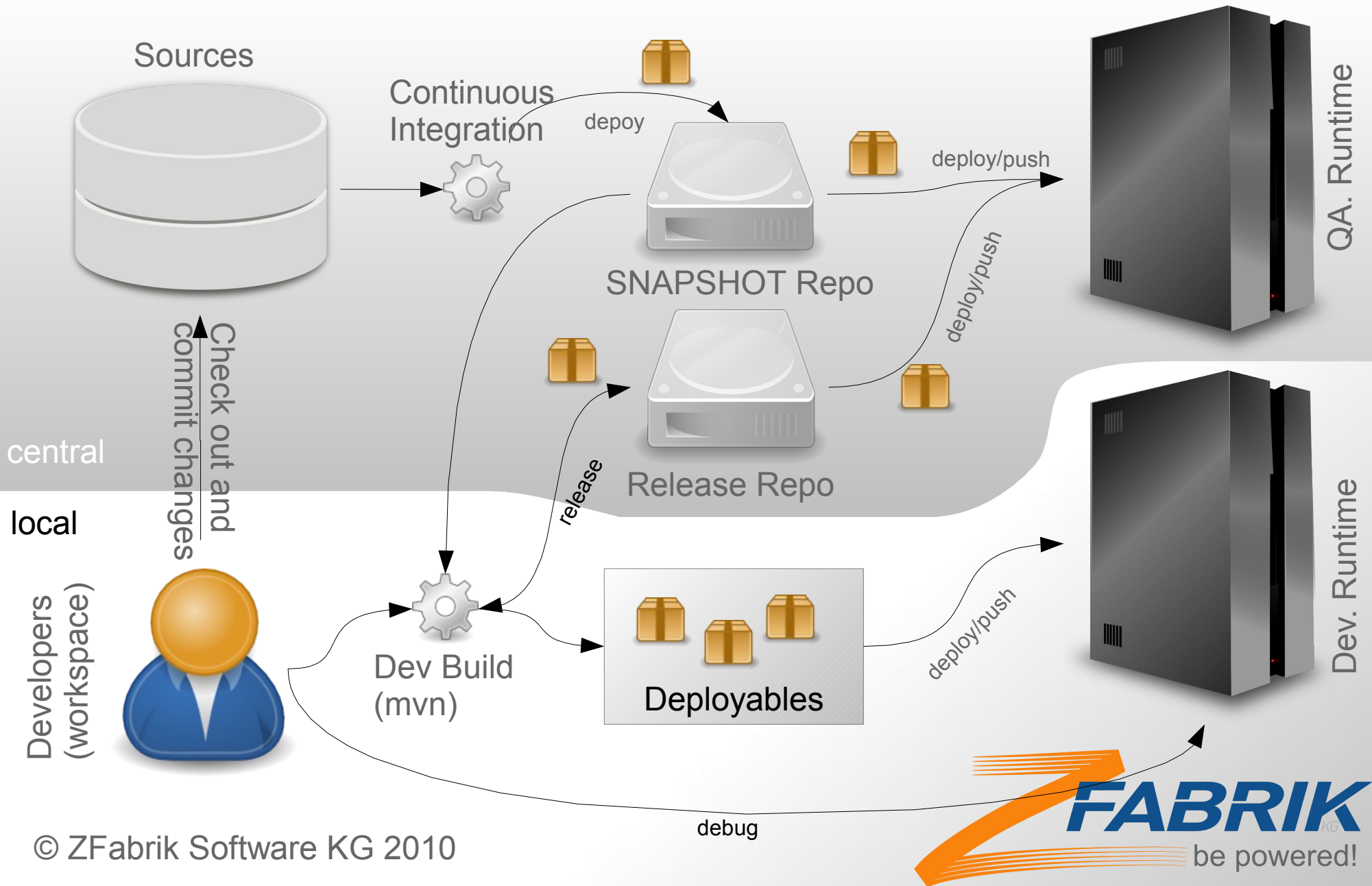
# The "classical" approach



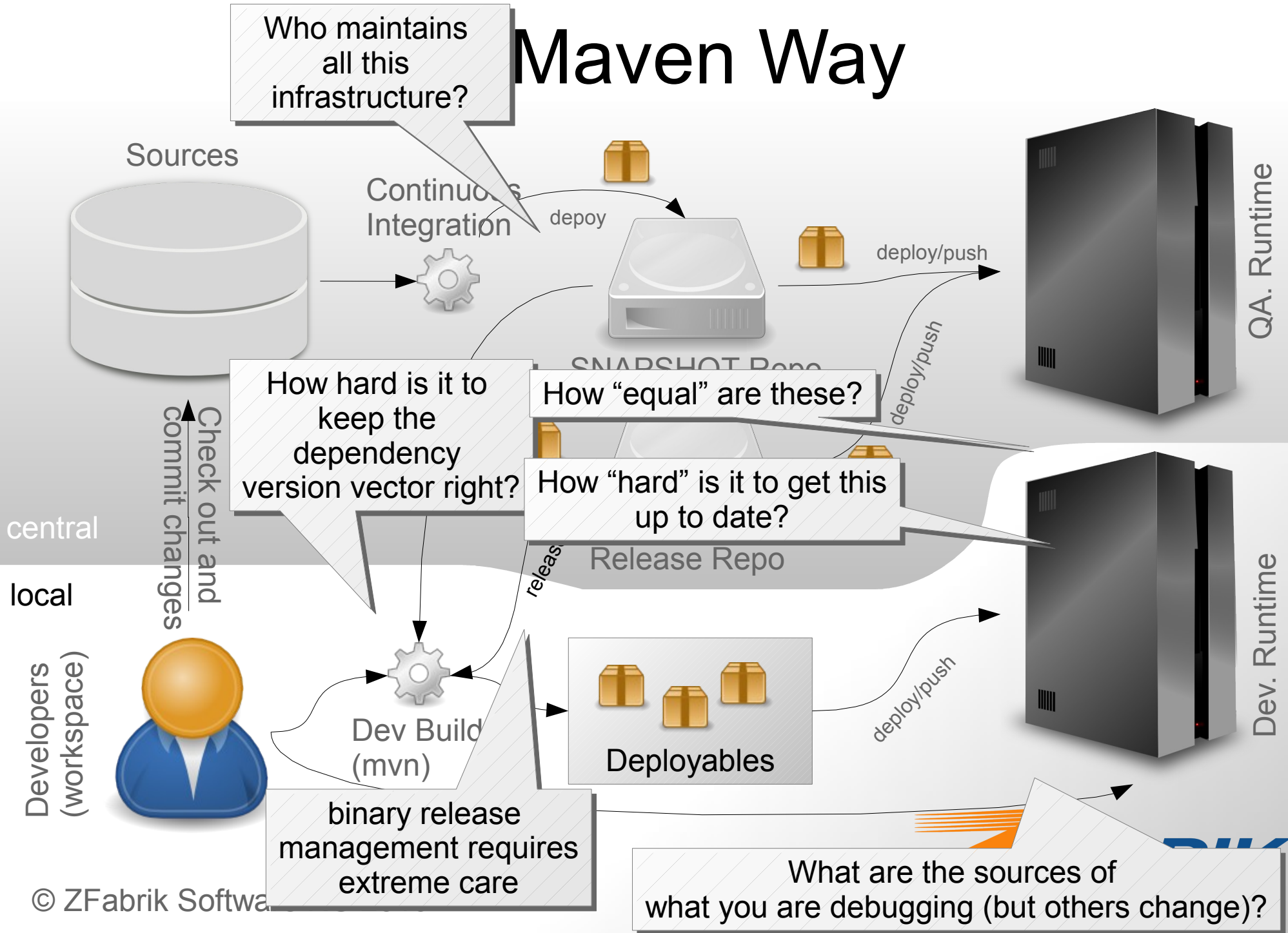
# The “classical” approach

- Ok for monolithic projects  
(only few deployables)
- Breaks for modular systems  
(too complex to keep in sync: Due to the “push” approach, it's up to the developer to keep things straight)
- Breaks for large projects  
(turn around times too long)
- **Summary:** Simple but doesn't scale.

# The Maven Way



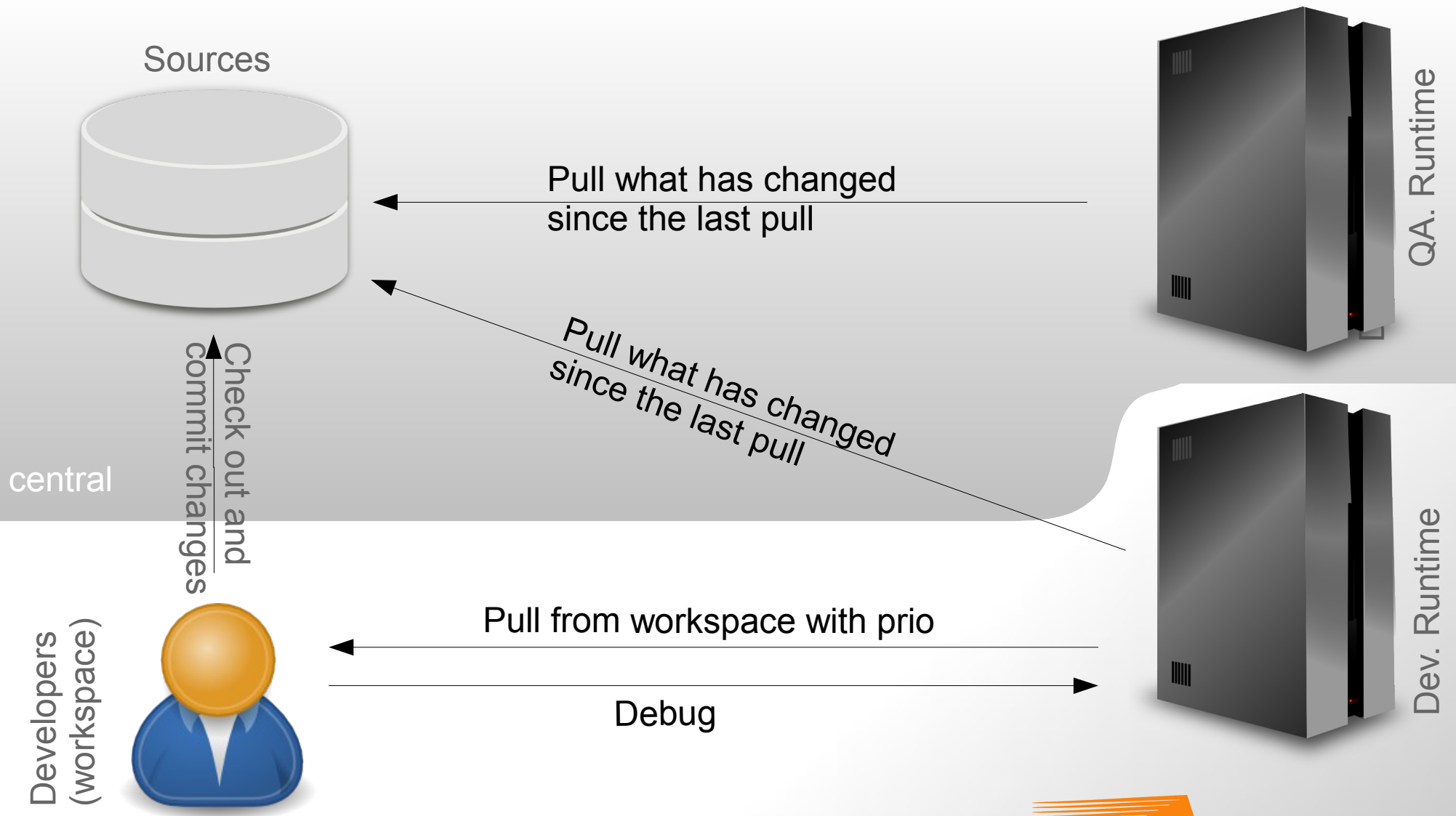
# Maven Way



# The Maven Way

- Maven is built around versioning, versioned dependency and release processes for binaries
- It does not address the system consistency side of things
- It is geared towards a distributed, independent community of producers of libraries
- It does not care about source consistency
- **Summary:** Useful for the community, much less interesting for solution providers.

# And in z<sup>2</sup>



# In z<sup>2</sup>

- A runtime that updates itself on-demand, according to changes in source repositories
- Developer runtime takes developer workspace into account with preference
- The runtime is a faithful representation of the repository – no question where sources are.
- Almost like a scripting environment

# Summary

- Standard Java development approaches do either not scale (classical approach) or do not have solution providers in mind
- The z<sup>2</sup> Environment is most useful for team development of solutions, rather than for providing libraries to the community

# More Information

Try it out at:

[www.z2-environment.de](http://www.z2-environment.de)

Contact us at:

[contact@zfabrik.de](mailto:contact@zfabrik.de)