



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Conclusions

Hosted by Alex Thalhammer

Conclusions of/for Atruvia 😊

- Enterprise architecture
 - Use NX monorepo with AA/DDD package
- Architecture solutions
 - Deployment monolith
 - Separate SPAs (Angular Apps or Angular Elements)
 - Shared Header / Footer
 - Includ. Header / Footer
 - Module Federated for MFE with AA/MF
 - Good for performance
 - Still a bit complicated -> might lead to issues

Conclusions of/for Atruvia 😊

- Use RxJS
 - Observables
 - And Subjects
- Use NgRx
 - Start using it now and no regrets on migrating later
 - Easy generation of boilerplate code (even with DDD package)
 - If you have no state in a feature → no need to invent one

Conclusions of/for Atruvia 😊

- Try to lazy load (almost) everything
 - Use intelligent preloading strategy
- Use ChangeDetection.OnPush as default
 - To get best runtime performance

Conclusions of/for Atruvia 😊

- Service worker can enhance your Angular App
- Quick Wins
 - WebApp Manifest
 - Offline-Modus
 - API Caching
 - Push Notifications
 - LocalStorage / IndexedDB



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Conclusions of/for Atruvia 😊

- Use ngx-translate for translations
- Use JEST & Cypress for testing
- Use prettier and ESLint
 - See some ESLint recommendations in our training repo
 - Add custom lint rules (e.g. ChangeDetection.OnPush)
- Use the new NG Dev Tools and Redux Inspector tools

Conclusions of/for Atruvia 😊

Feedback

- What should be improved?
- Feedback-URL: <https://forms.gle/iWf4RTKk8rhtdZQy8>

Thank you!



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT