



DSLs for Application Development

Tyler Hodgkins, YP Language Tooling

Safe Harbor Statement

This presentation may contain forward-looking statements for which there are risks, uncertainties, and assumptions. If the risks materialize or assumptions prove incorrect, Workday's business results and directions could differ materially from results implied by the forward-looking statements. Forward-looking statements include any statements regarding strategies or plans for future operations; any statements concerning new features, enhancements or upgrades to our existing applications or plans for future applications; and any statements of belief. Further information on risks that could affect Workday's results is included in our filings with the Securities and Exchange Commission which are available on the Workday investor relations webpage: www.workday.com/company/investor_relations.php

Workday assumes no obligation for and does not intend to update any forward-looking statements. Any unreleased services, features, functionality or enhancements referenced in any Workday document, roadmap, blog, our website, press release or public statement that are not currently available are subject to change at Workday's discretion and may not be delivered as planned or at all.

Customers who purchase Workday, Inc. services should make their purchase decisions upon services, features, and functions that are currently available.

About Us

- Est. 2005
- ~7000 employees



- HR
- Financial
- Payroll
- Student Services
- Recruiting
- Analytics
- **Developer Platform**

XpressO

- Object-Oriented
- Functional
- Reflective
- Metadata
- Dynamic

Featuring:

- Compensation
- Consolidations
- Payroll
- **Allocations**
- Revaluation
- Planning
- Budgeting
- Reporting
- Recruiting
- Data Science
- Settlements
- Tax
- Revaluation
- Procurement
- Business Intelligence
- Student services
- Employee management
- Organizations
- Projects
- Awards

XpressO DSLs

- UI: Tasks and Elements
- APIs: REST and High-volume web services
- Condition Rules
- Business Process Definitions
- Reporting
- ... and plenty more



XpressO

Issues

- Slow functional development
- 10 years of haphazard DSL development
- 2000 active developers

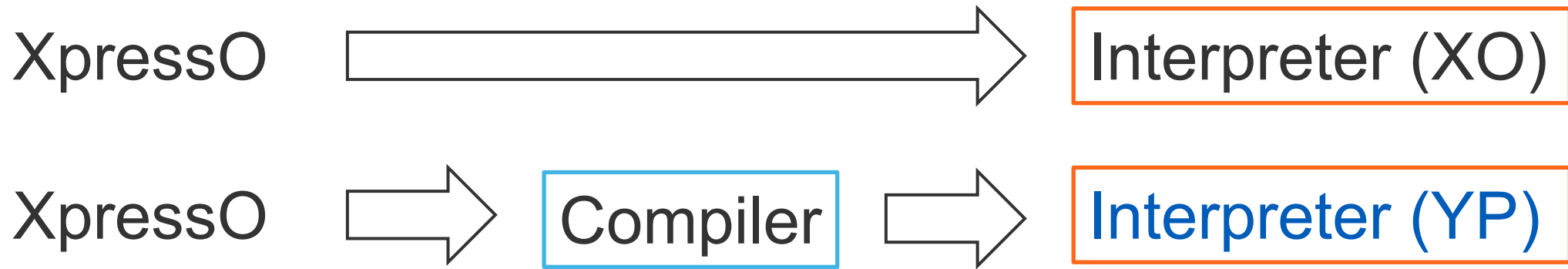
Challenges

- Building from scratch takes a long time
- Massive existing codebase mandates interoperability
- Existing tooling likely won't be compatible
- Where to start?

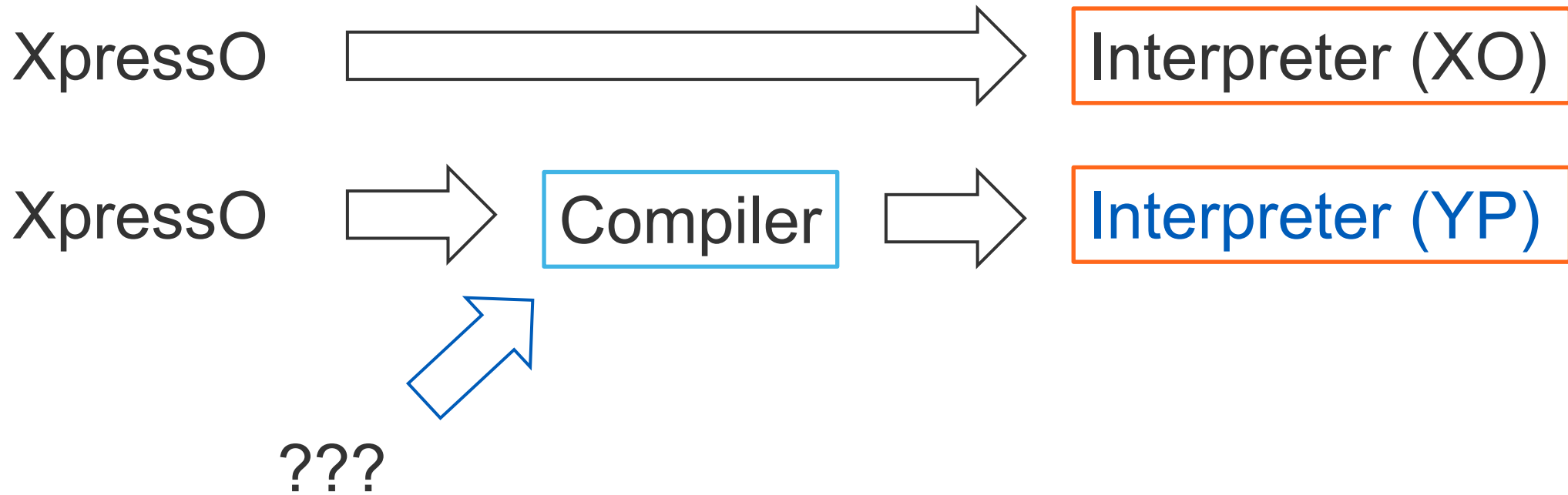
Execution



Execution



Execution





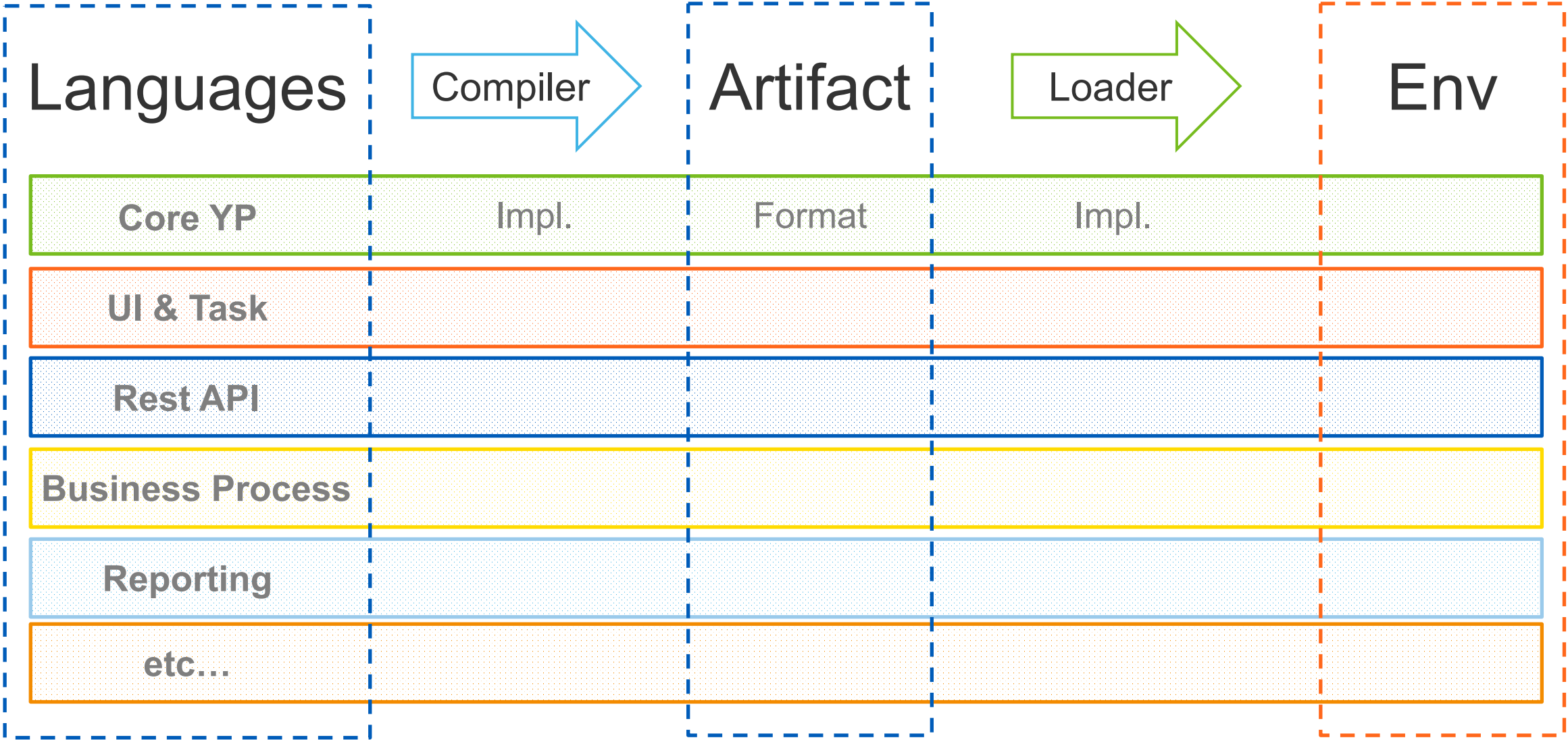
YP Language

Code Lifecycle



- Compiler transforms to common artifact (.yar)
- Loader handles many artifacts
 - Resolves references
 - Generates necessary runtime information
- How can we handle many teams building different DSLs?

Language Platform



Conclusions

- DSLs are excellent in practice
- Useful for large-scale development
- Particularly hard to maintain if not designed for flexibility
- Modern tools are extremely powerful, facilitating even fundamental, large-scale migrations

“Research Questions” (Tasks)

- Will a language platform scale?
- **Versioning**
- Can we do this in the cloud?



Q&A