SFDX Intro

Making Salesforce development more accessible to more developers

ISV Partner Problems

- Tools that Developers can use without knowing Salesforce
- Developer Orgs that proliferate as developer teams change
- Source Control
- Must deploy over multiple client orgs and configurations

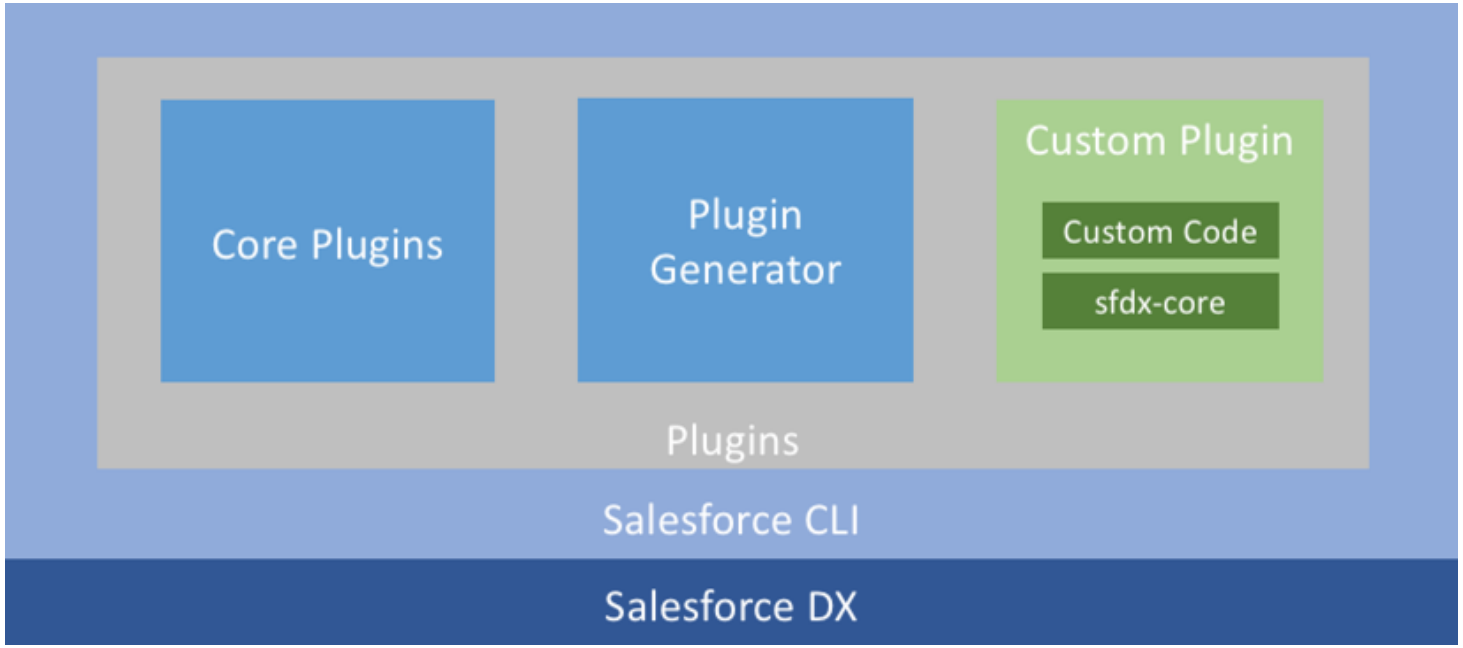
Customer Problems

- Tools that Advanced Admins can also work with
- Monitoring vendor work in progress
- Maintaining data security
- Strong UAT efforts
- Maintainging security of Intellectual Property
- Source Control / Metadata backup
- Must maintain uniform compliance and configuration

SFDX Solution

- Familiar developer tools like VS Code
- Support for **Sandbox** development for Customers
- Support for Scratch Org development for ISVs

SFDX Flexibility and Familiarity with VS Code
IntelliJ
Welkins Suite



Customer Orgs are Not ISV Orgs

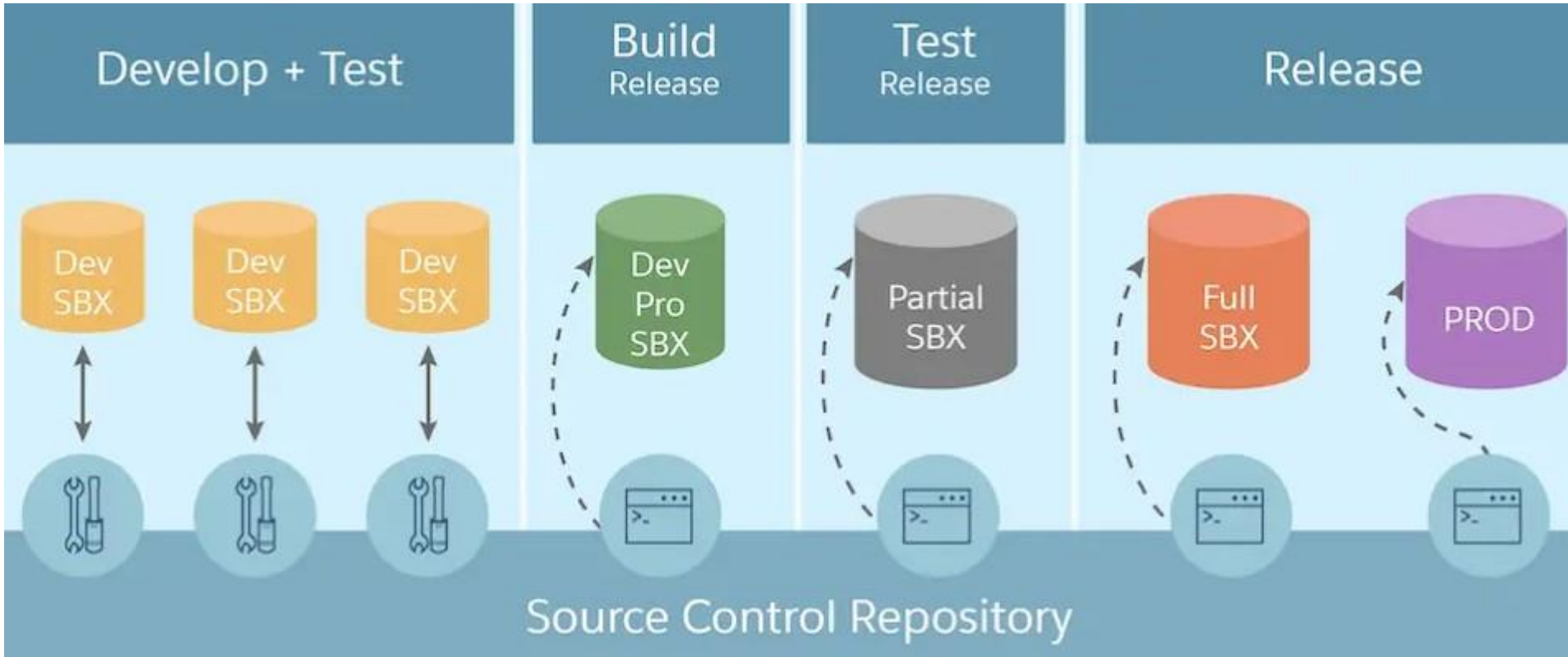
- Our apps and packages run like embedded systems
- Solutions may encapsulate their functionality
- Each part must work as part of the whole

Advantages of Sandboxes, a Holistic Approach

- Testing new development on the platform where it will live
- Collaboration with Admins and Devs
- Easy to freeze users and delete orgs
- Flexibility of Packages with dependencies and Packages without dependencies all defined in the org
- Easy to audit and monitor
- Works with existing code
- Supports our existing isSandbox code

Package-Based Development Includes Change Sets and Packages Together

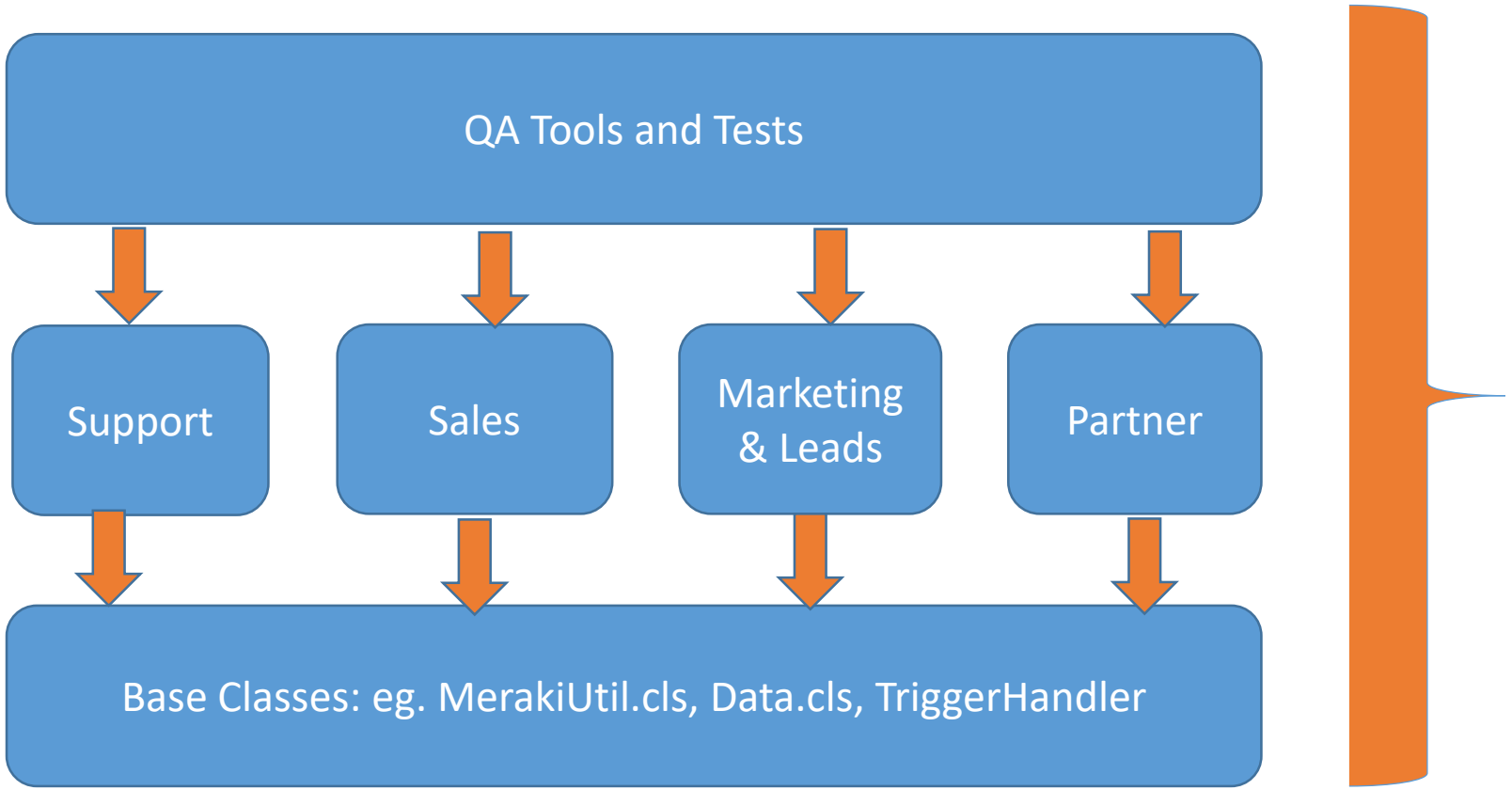
- Un-deployed change sets are mutable, so the feature package can grow and change over time
- Package.xml defined via GUI
- Familiar tools with new SFDX features



More Awesome Options with Sandboxes

- Follows best practices regarding the software development life cycle. It's compatible with the new features of Salesforce DX: projects, source-driven development commands.
- Encapsulates all the changes you are tracking between life cycle stages in a versioned artifact.
- Makes it easier for you to accommodate new feature requests. Simply add, update, and remove components in your package as defined in the GUI of your Sandbox or Prod org.
- Provides an improved audit history, so you can more easily track and understand the changes made to your production org.
- Organizes source. It's much easier to know which components belong to which applications and features.
- Promotes iterative and modular development.
- Supports interdependencies.
- Supports continuous integration and continuous delivery because the packaging CLI commands enable each step in the deployment pipeline to be fully automated.

Multiple Repository Approach with Packages



Maintained as one platform with Sandboxes