App Modernization –
Migration approaches

- WW Developer Advocate Team

IBM Developer

Moving apps to modern architectures starts with planning

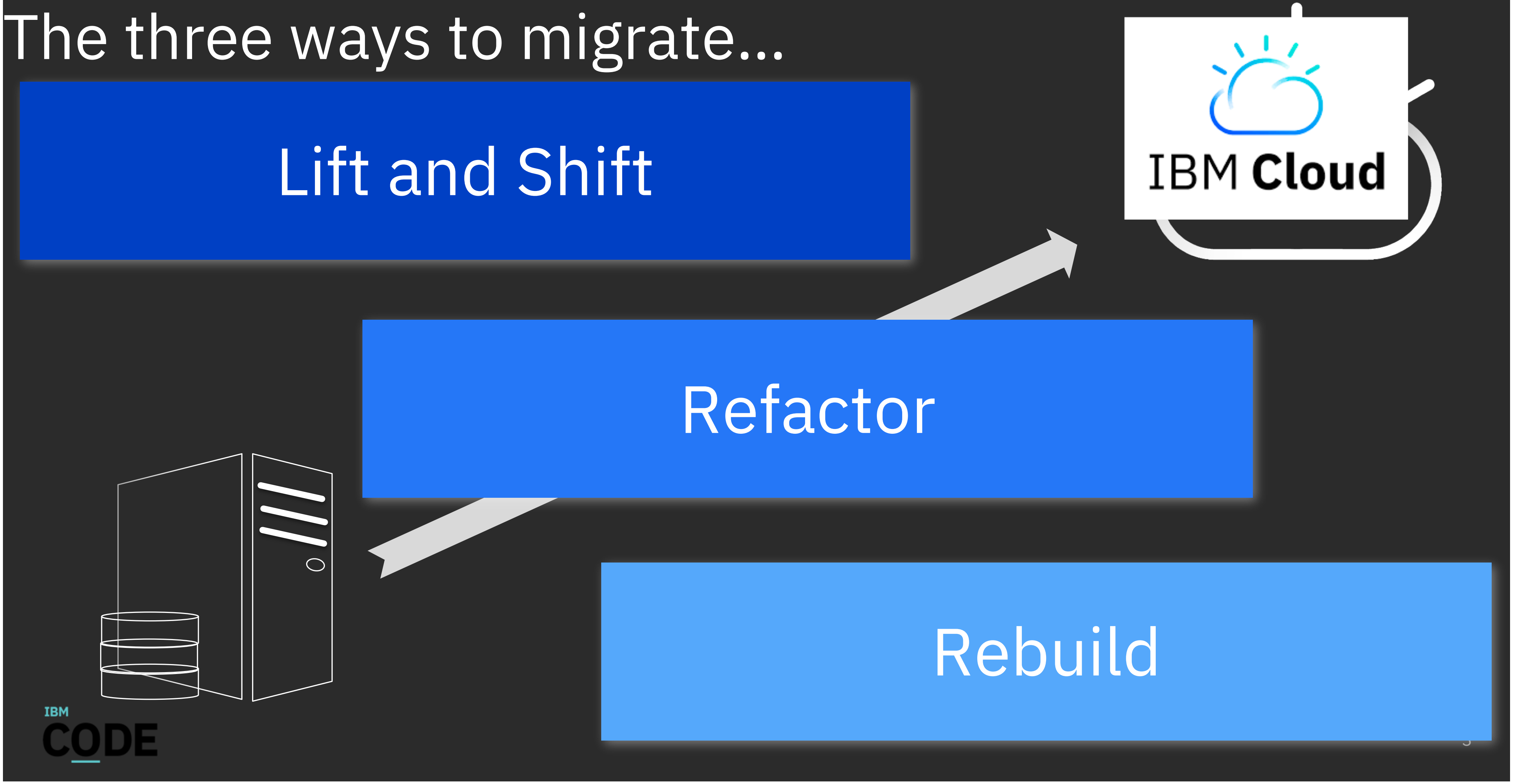
For each legacy app the following need to be assessed:

i. Business value
  - Measure cost of migration vs benefits gained by migrating
    - Good value usually found in apps that have
      - Multiple upgrades per year
      - Lots of business rules requiring complex regression testing and extended service outages to implement
      - Large, stable (or growing) user base

ii. Technical effort
  - This is where Transformer Advisor helps

Best candidates are those with relatively high business value and relatively small technical effort

The three ways to migrate...



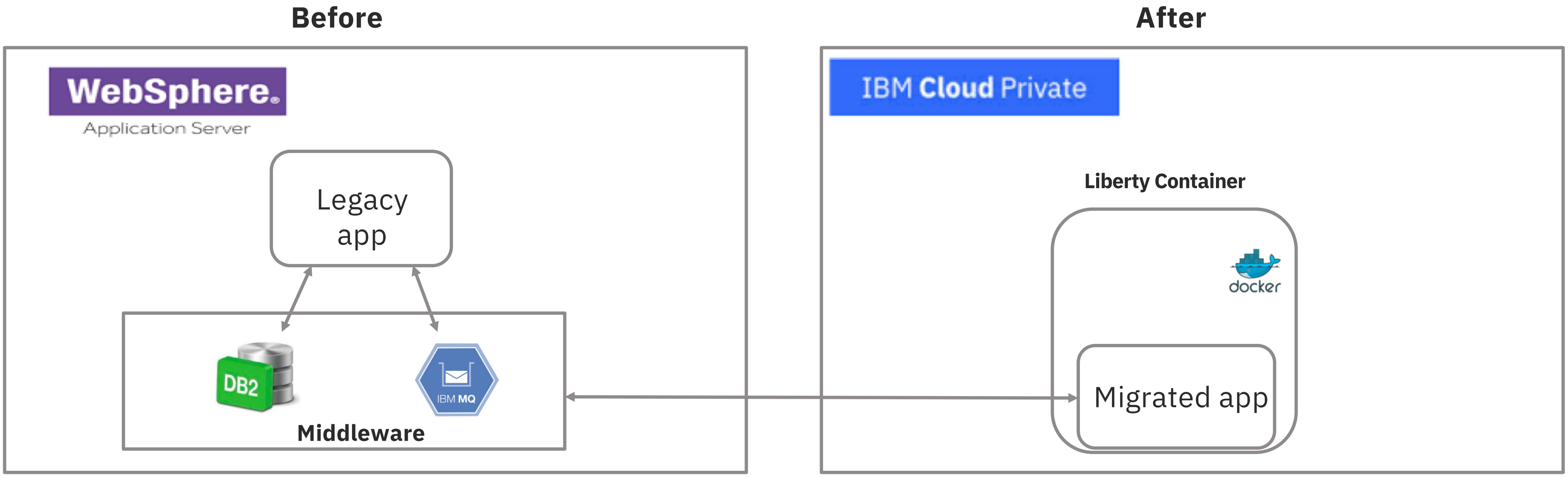
Lift and Shift

Refactor

Rebuild

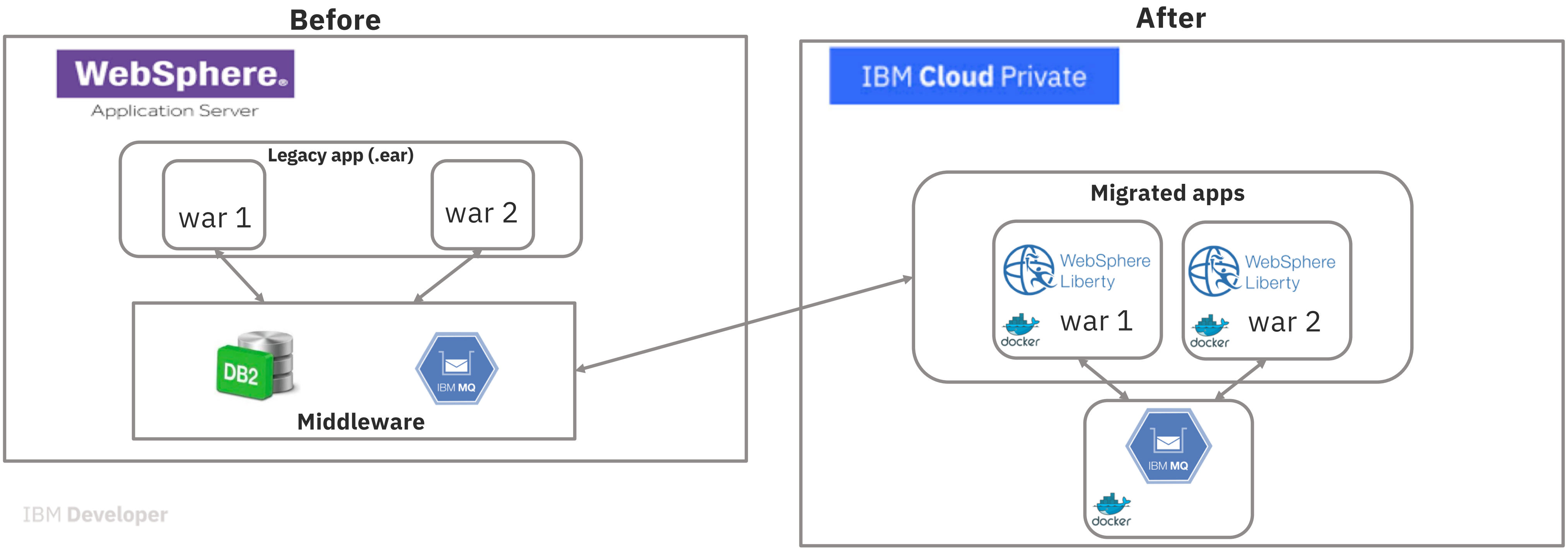
Lift and shift example

- Legacy app migrated in full from WebSphere Application Server Traditional to WebSphere Liberty running in a Kubernetes cluster
- Transformation Advisor provides guidance with equivalent Liberty config for migrated app
- Associated middleware is left in place and still used by migrated app



Refactor example

- Components (.war files) of legacy app (.ear) migrated to separate Liberty containers
- DB2 is left in place and still used by migrated app
- IBM MQ is partitioned
  - Container version handles communications between migrated modules
  - Legacy version handles communications between migrated modules and other legacy apps



Before

WebSphere Application Server

Legacy app (.ear)

war 1

war 2

Middleware

After

IBM Cloud Private

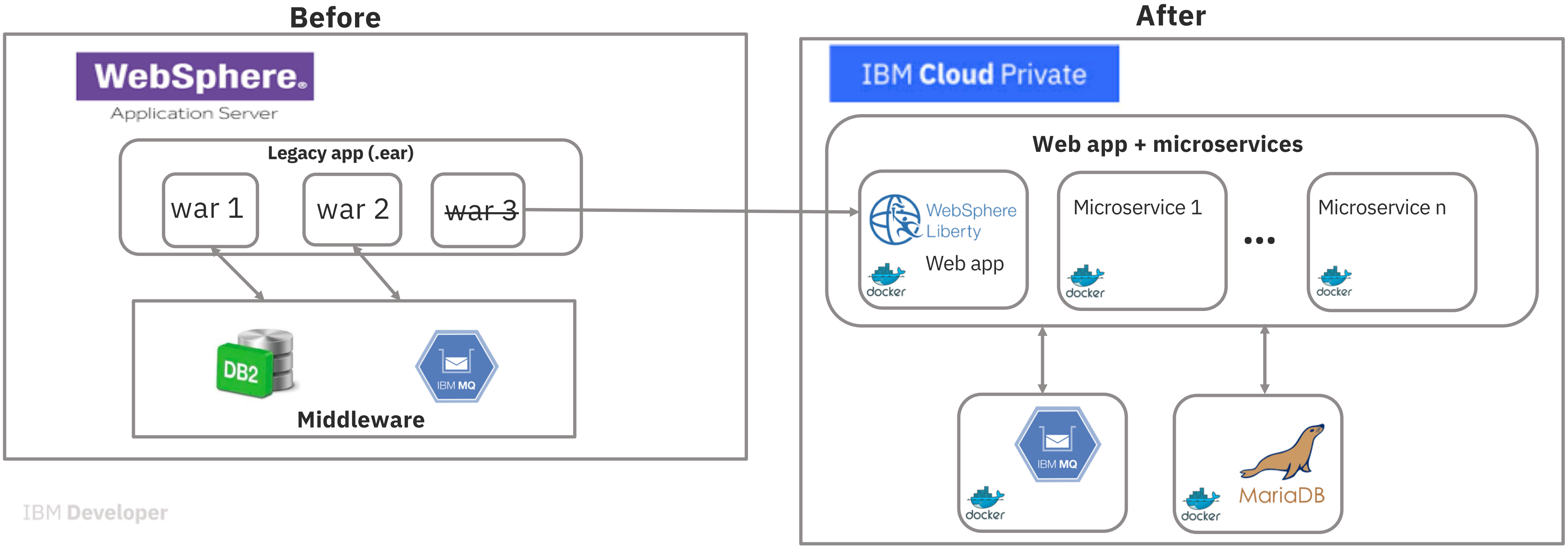
Migrated apps

WebSphere Liberty war 1

WebSphere Liberty war 2

Rebuild example using the Strangler Pattern

- Component (war 3) of legacy app is rebuilt as a web app and a set of microservices
- New microservices and app use their own containerized data and messaging middleware
- Legacy app proxies all requests for war 3 to new implementation and handles everything else
- Over time more of the legacy app is rebuilt and eventually it will be no longer be needed



References

- Newman, Sam. *Building Microservices: Designing Fine-grained Systems*. S.l.: O'Reilly Media, Inc, USA, 2018.
  - *Chapter 5 – Splitting the Monolith* specifically covers rebuilding a legacy app as a set of microservices
- Evans, Eric. *Domain-driven Design: Tackling Complexity in the Heart of Software*. Boston, MA: Addison-Wesley, 2014.
  - *Chapter 14 – Maintaining Model Integrity* defines a *bounded context*. Identifying bounded contexts is the first step in splitting a monolithic application.
- BoundedContext – *Martin Fowler*
  - https://martinfowler.com/bliki/BoundedContext.html
- Feathers, Michael C. *Working Effectively with Legacy Code*. Upper Saddle River, N.J: Prentice Hall PTR, 2013.
  - *Chapter 4 – The Seam Model* defines a *seam*. Identifying seams with the appropriate level of granularity is an important step in defining microservice boundaries
- Apply the Strangler Application pattern to microservices applications – *Kyle Brown IBM Fellow*
  - https://www.ibm.com/developerworks/cloud/library/cl-strangler-application-pattern-microservices-apps-trs/index.html?ca=drs-
- Transformation Advisor
  - https://developer.ibm.com/recipes/tutorials/using-the-transformation-advisor-on-ibm-cloud-private/
- Introduction to IBM's app modernization approach
  - https://www.ibm.com/cloud/garage/content/culture/app-modernization-field-guide/