

Records in Action



Roland Guijt

Microsoft MVP, Consultant, Trainer

@rolandguijt rolandguijt.com



Overview



From classes to records

Destructuring

Cloning

Custom record types

Records and inheritance



The Effects of “with”

Memory

classroomCourse



**Name = “Working with C# Records”
Author = “Roland Guijt”**

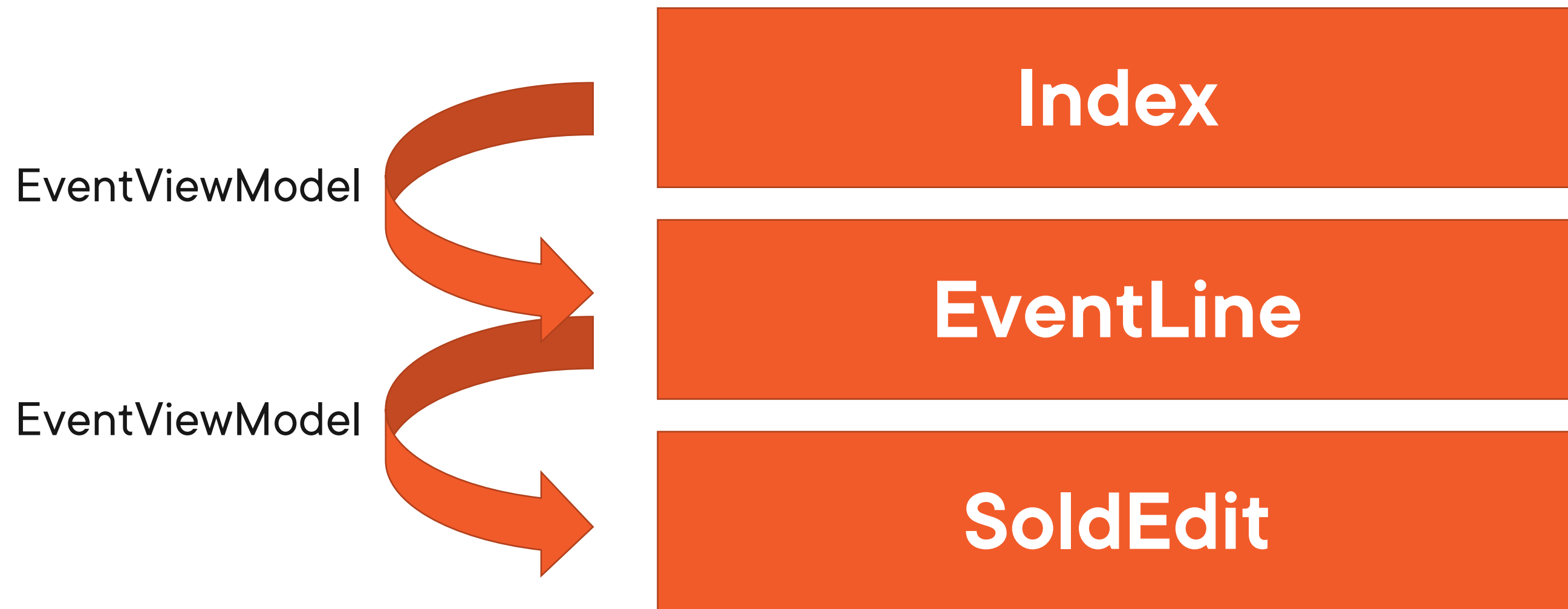
anotherCourse



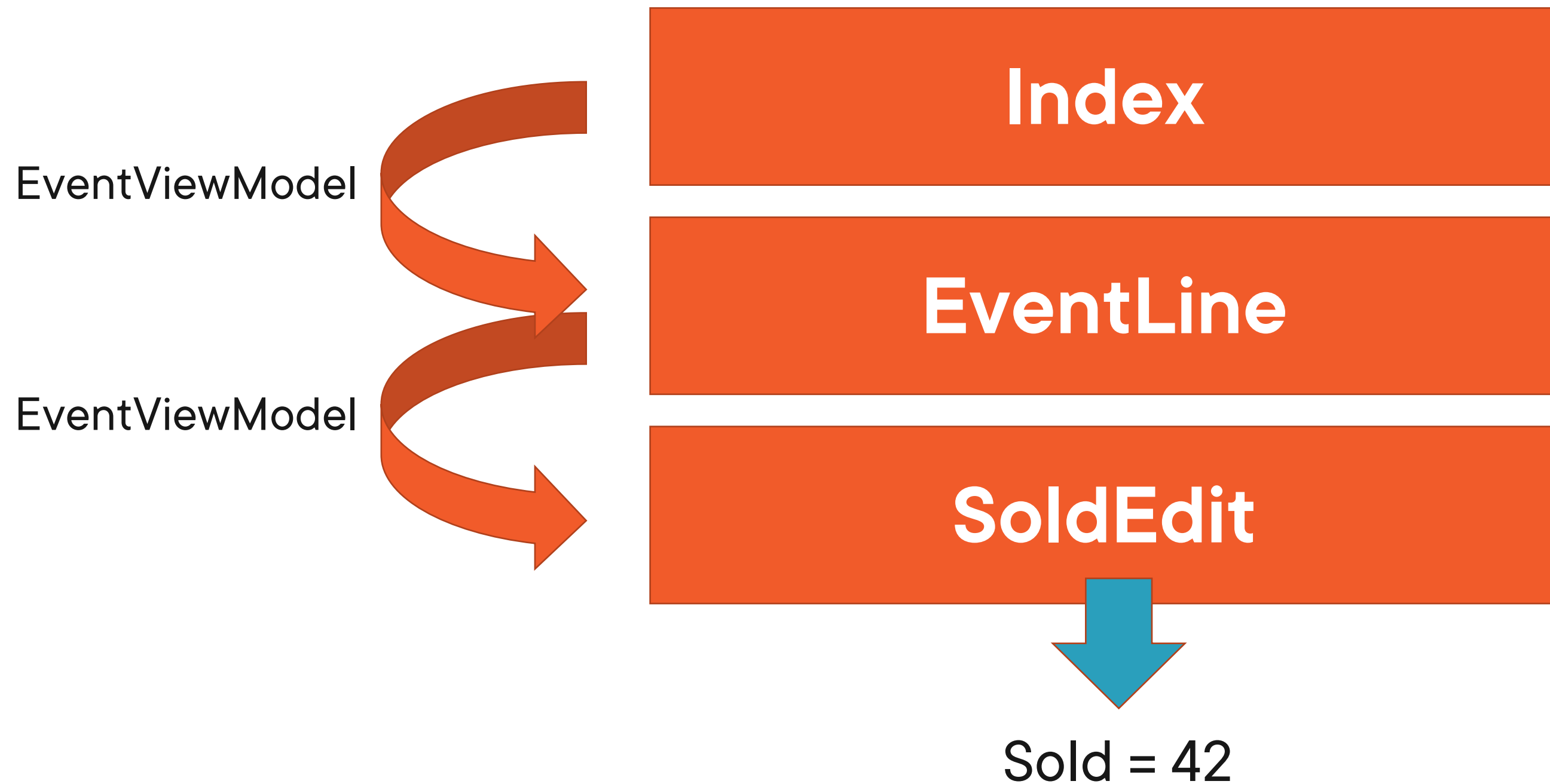
**Name = “C# Getting Started”
Author = “Roland Guijt”**



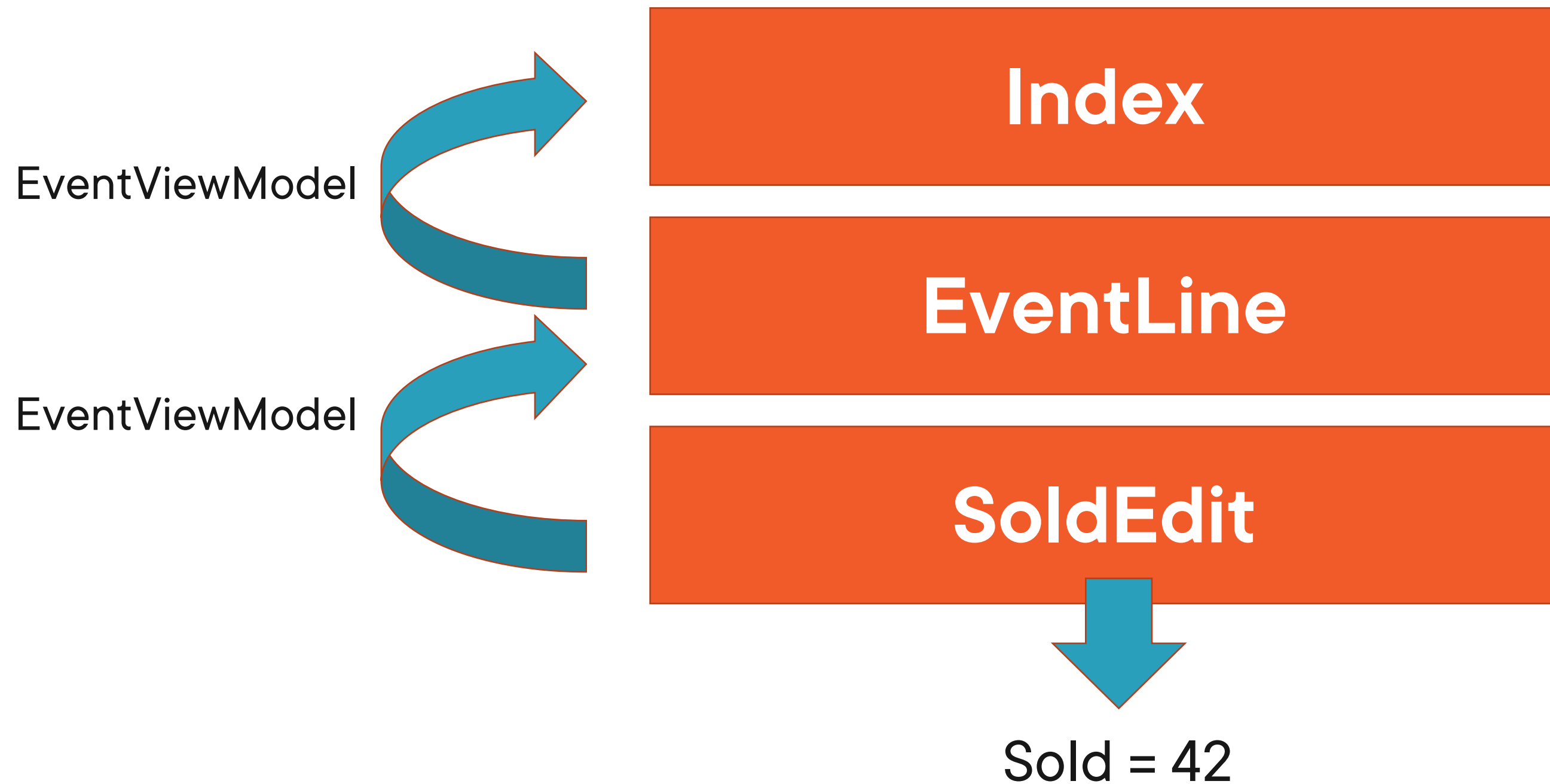
GloboTicket Application Components



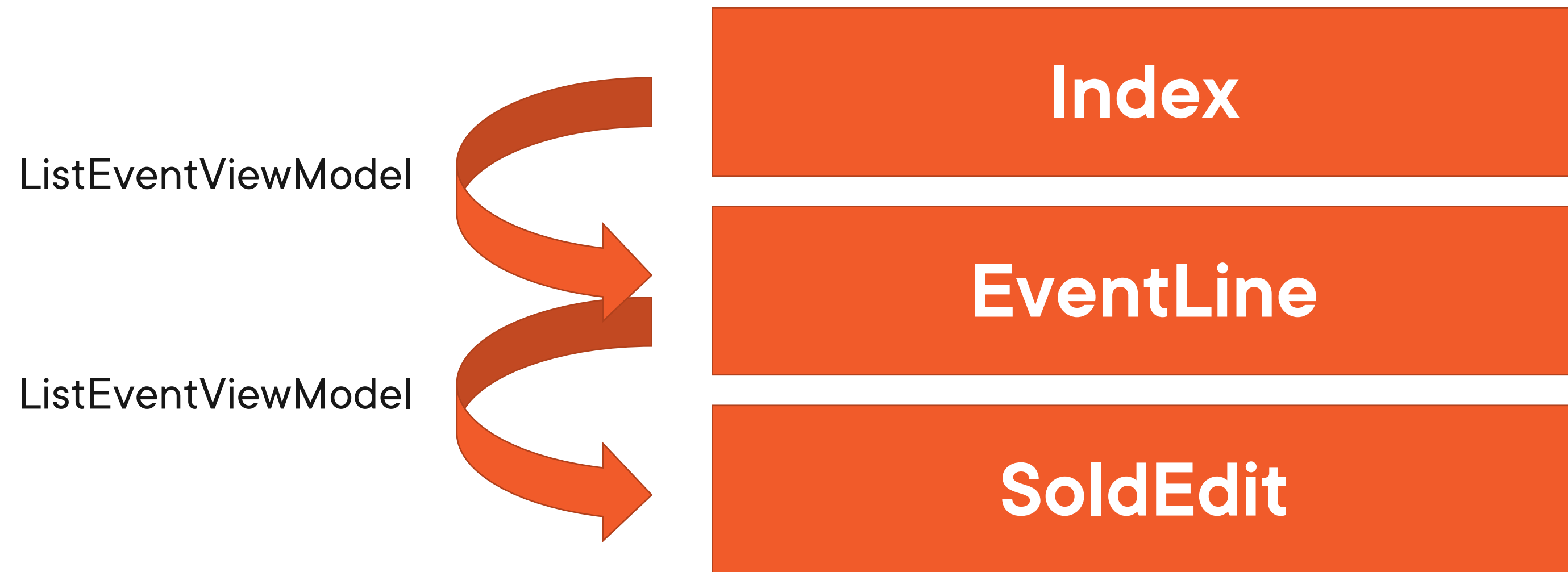
GloboTicket Application Components



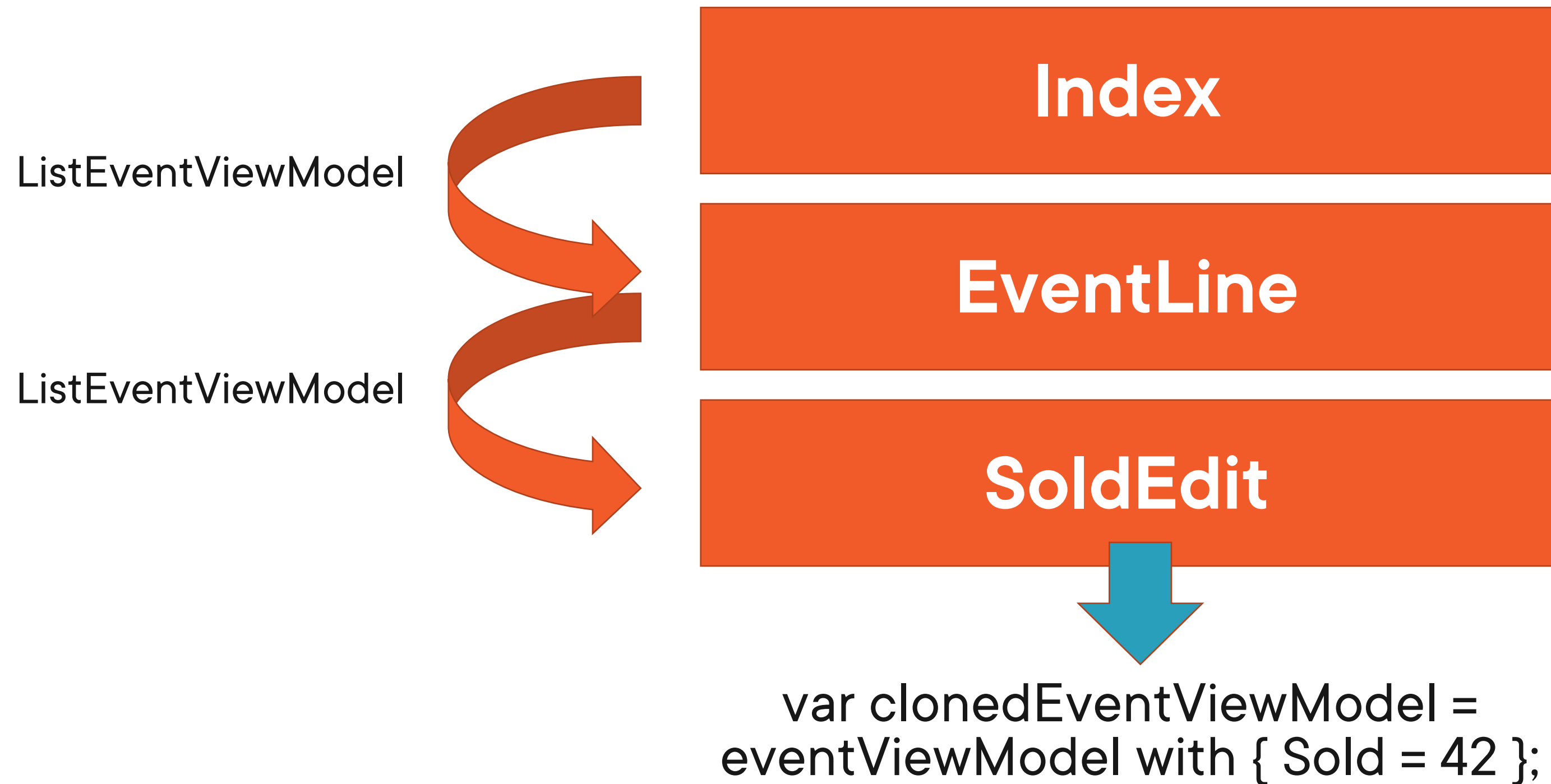
GloboTicket Application Components



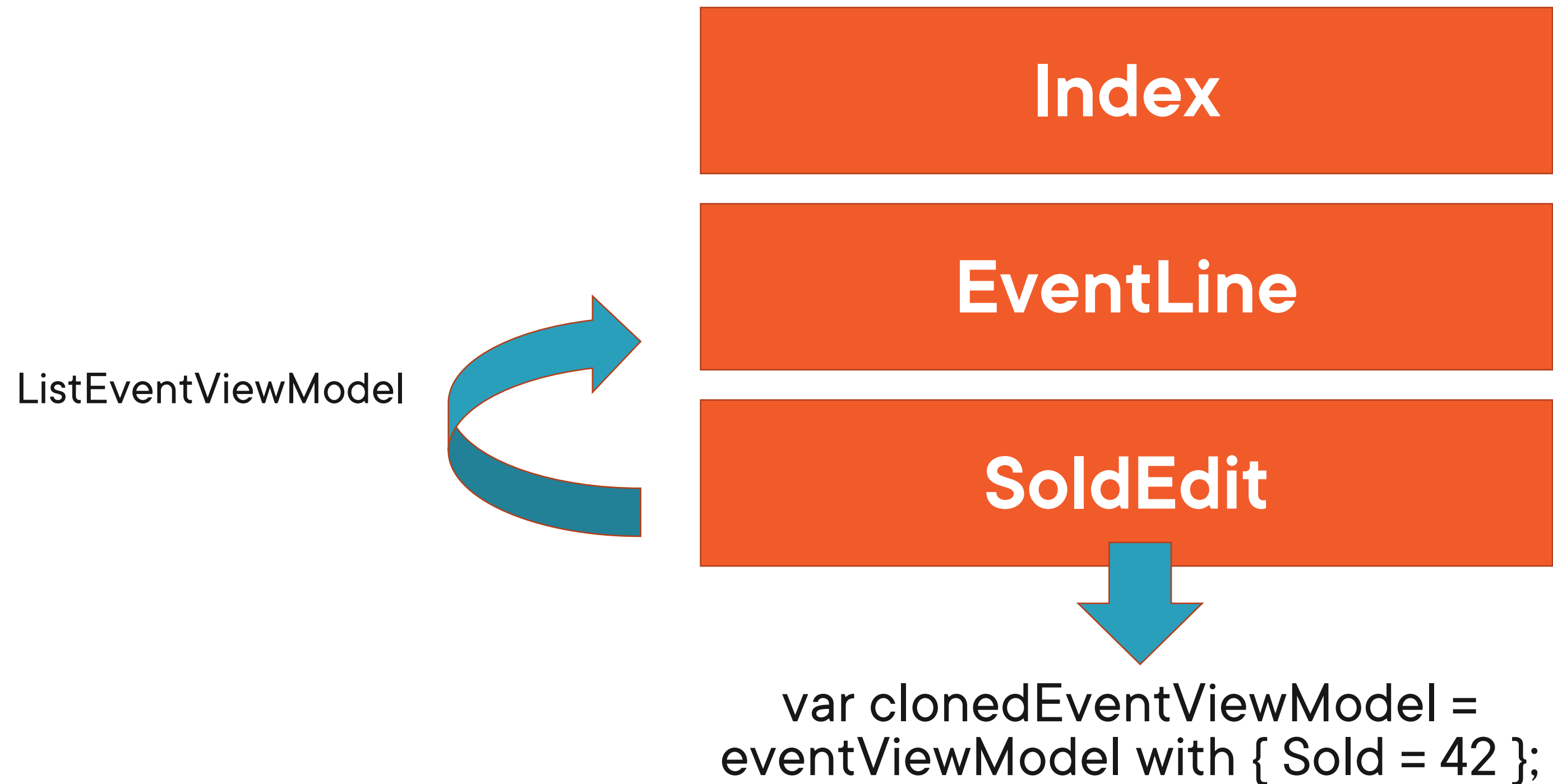
GloboTicket Application Components



GloboTicket Application Components



GloboTicket Application Components



Custom Records

Records can have bodies containing everything a class can have in its body

Constructor and deconstructor is only generated for positional properties

A mix of positional and properties in the body is possible

Make sure to use init as the setter

Init can also be used in classes



Inheriting from Records

Records can only derive from other records

**Derived positional records have to repeat all
property declarations**



Summary



Using records instead of classes can solve (potential) problems

Positional records get a constructor and deconstructor for free

Records are reference types that can be easily cloned

Records can contain bodies and positional syntax can be used in conjunction

Inheritance is largely the same as with classes but keep the generated constructor in mind

