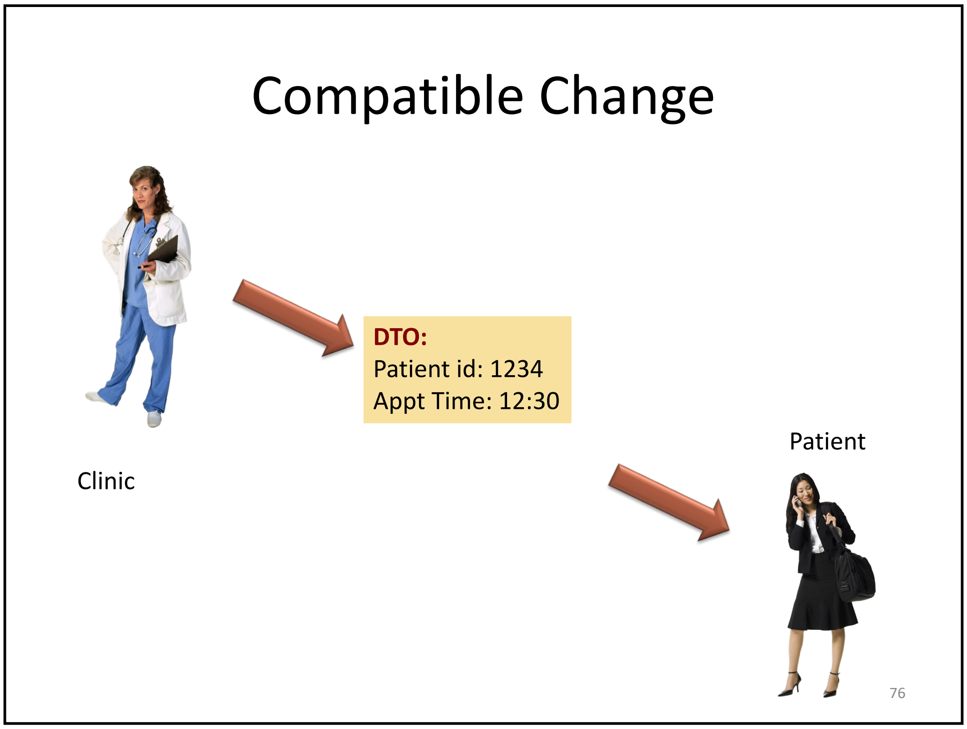
COMPATIBLE CHANGE: DATA CONTRACT

Compatible Change

- The service may want to upgrade
- Changes that don't break contract
  - Compatible with existing clients?
- Data contract
  - Data (DTO) from service to client
  - Data from client to service
- Operations contract
  - Operations provided by service to client
  - Callback operations provided by client to service

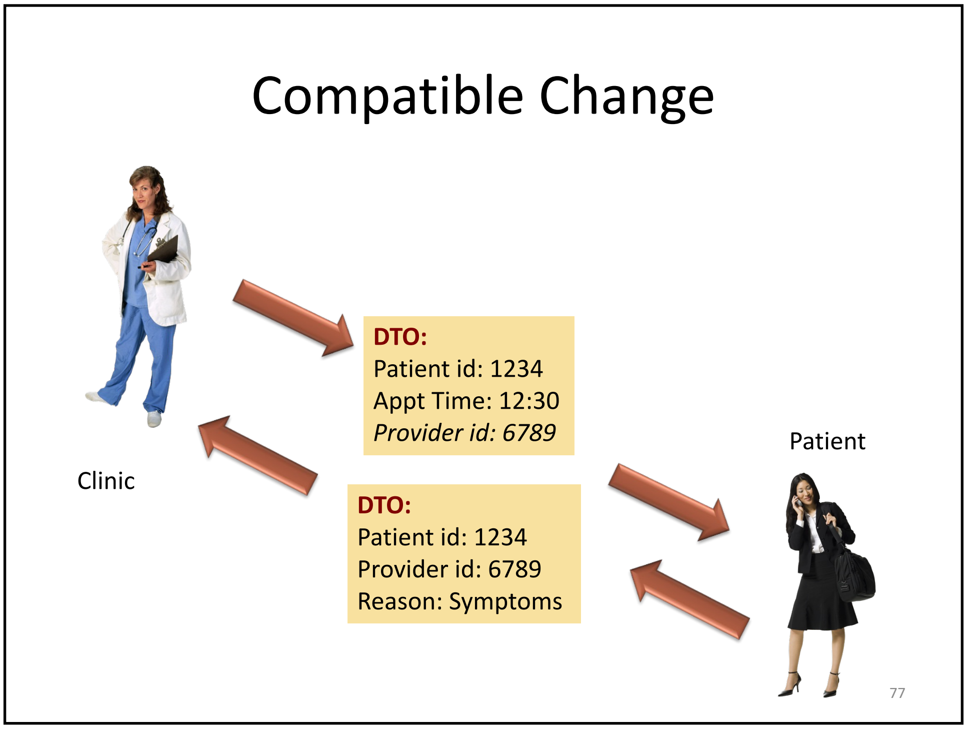


Compatible Change

DTO:
Patient id: 1234
Appt Time: 12:30

Patient

Clinic



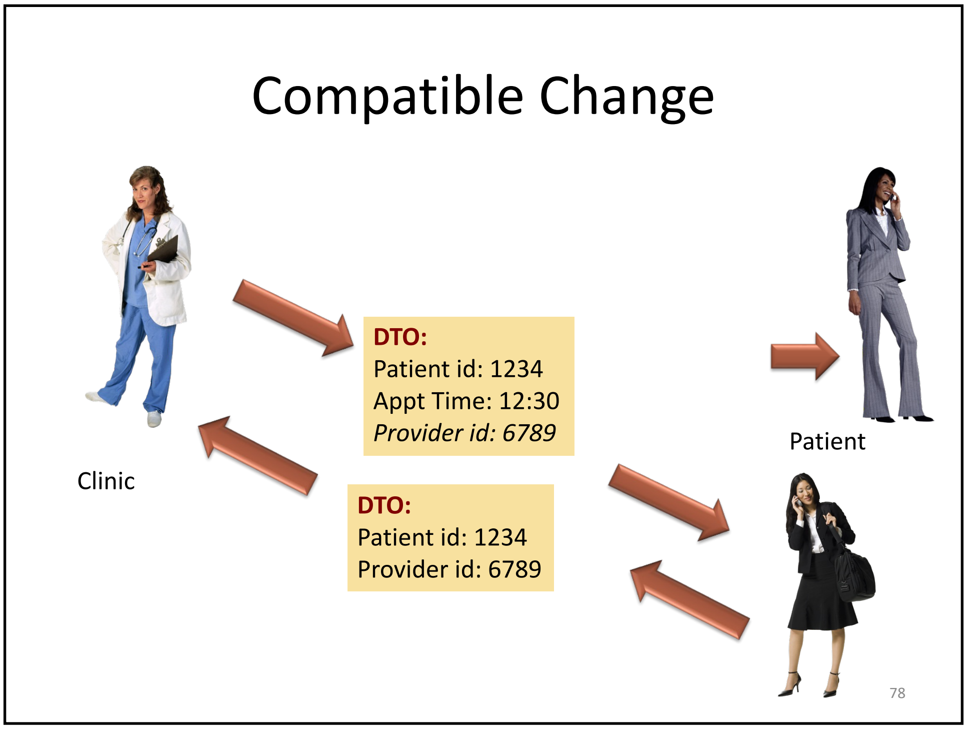
Compatible Change

DTO:
Patient id: 1234
Appt Time: 12:30
Provider id: 6789

Patient

Clinic

DTO:
Patient id: 1234
Provider id: 6789
Reason: Symptoms



Compatible Change

Clinic

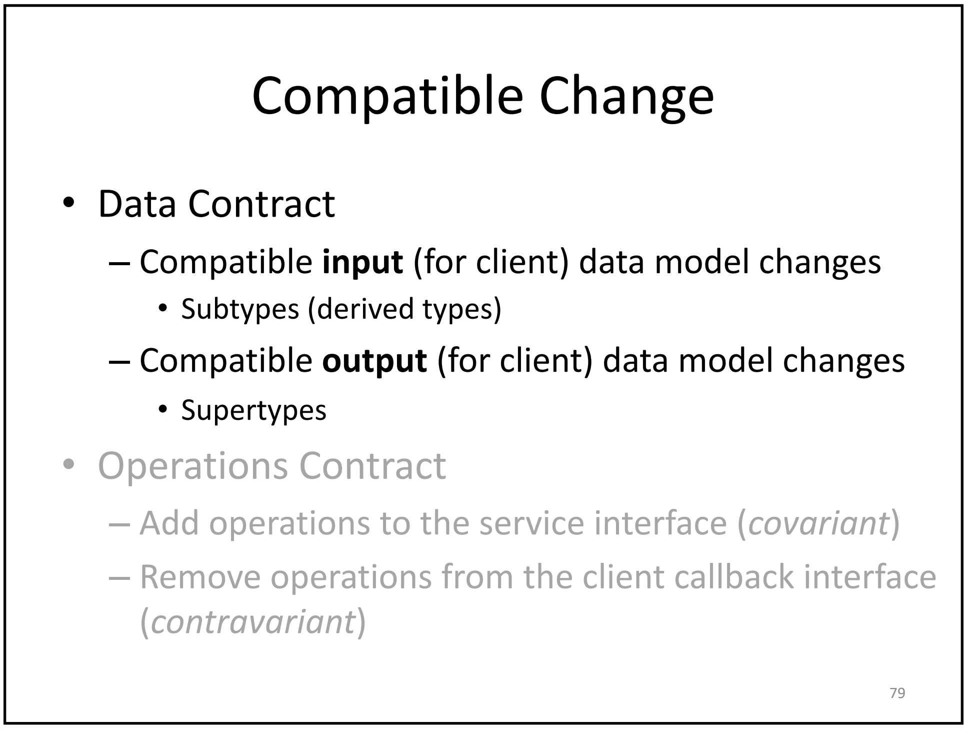
DTO:
Patient id: 1234
Appt Time: 12:30
Provider id: 6789

DTO:
Patient id: 1234
Provider id: 6789

Patient

78

78

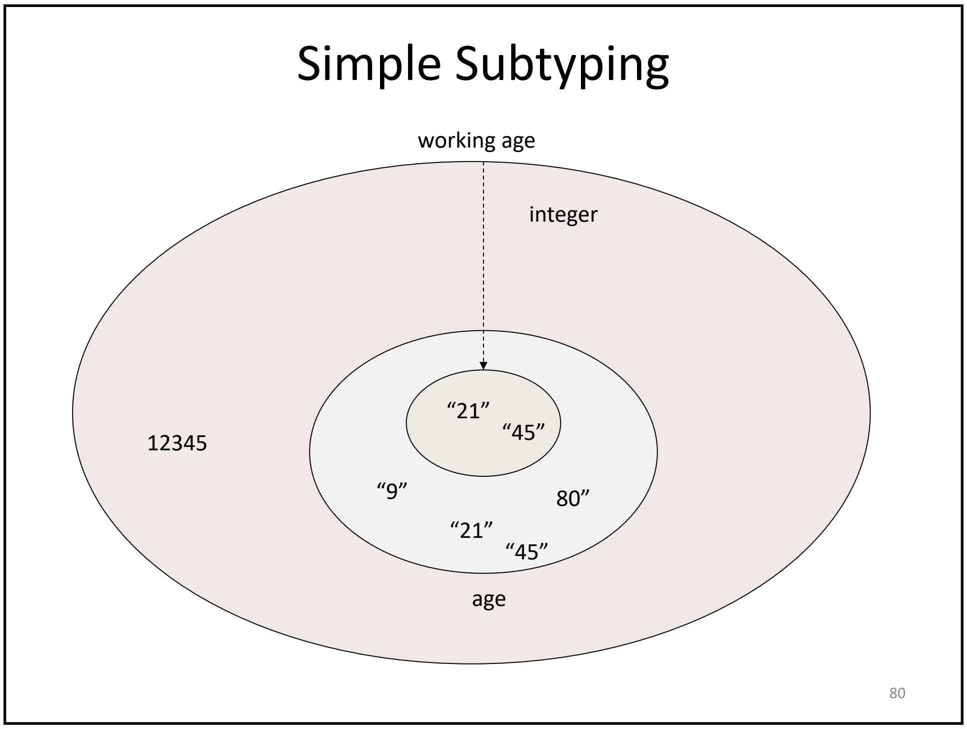


Compatible Change

- Data Contract
  - Compatible **input** (for client) data model changes
    - Subtypes (derived types)
  - Compatible **output** (for client) data model changes
    - Supertypes
- Operations Contract
  - Add operations to the service interface (*covariant*)
  - Remove operations from the client callback interface (*contravariant*)

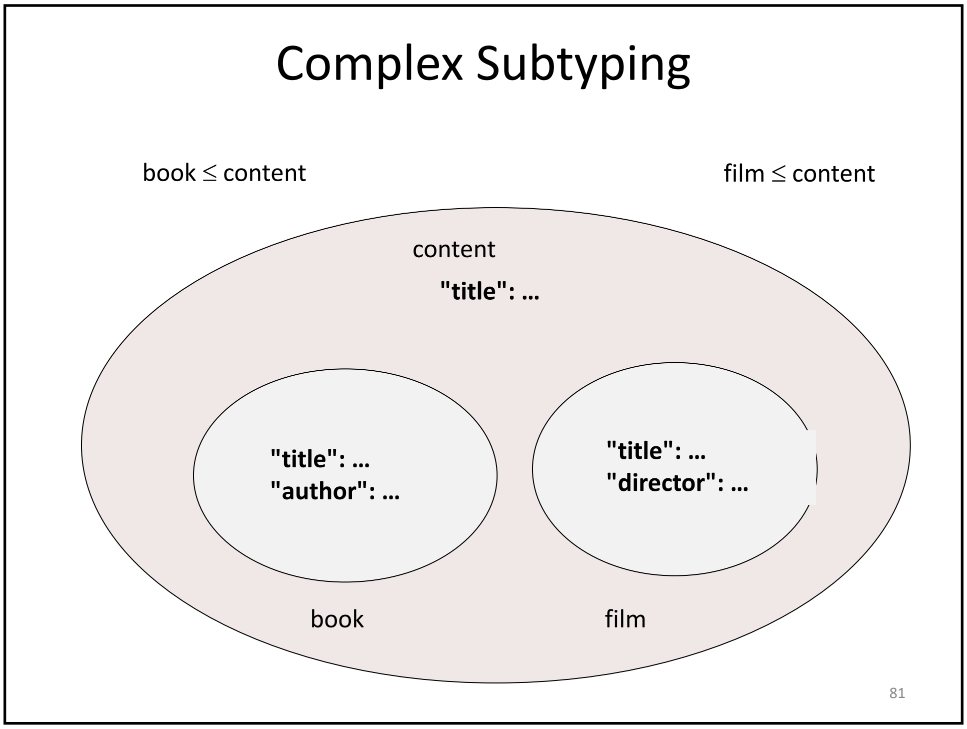
79

79



Simple Subtyping

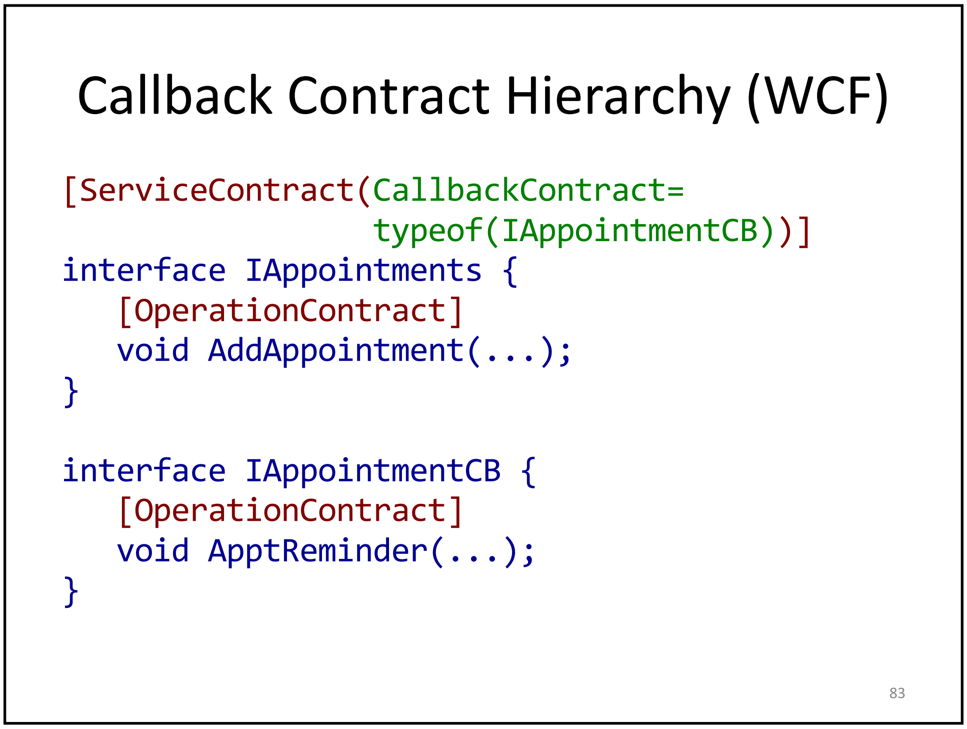
80



Complex Subtyping

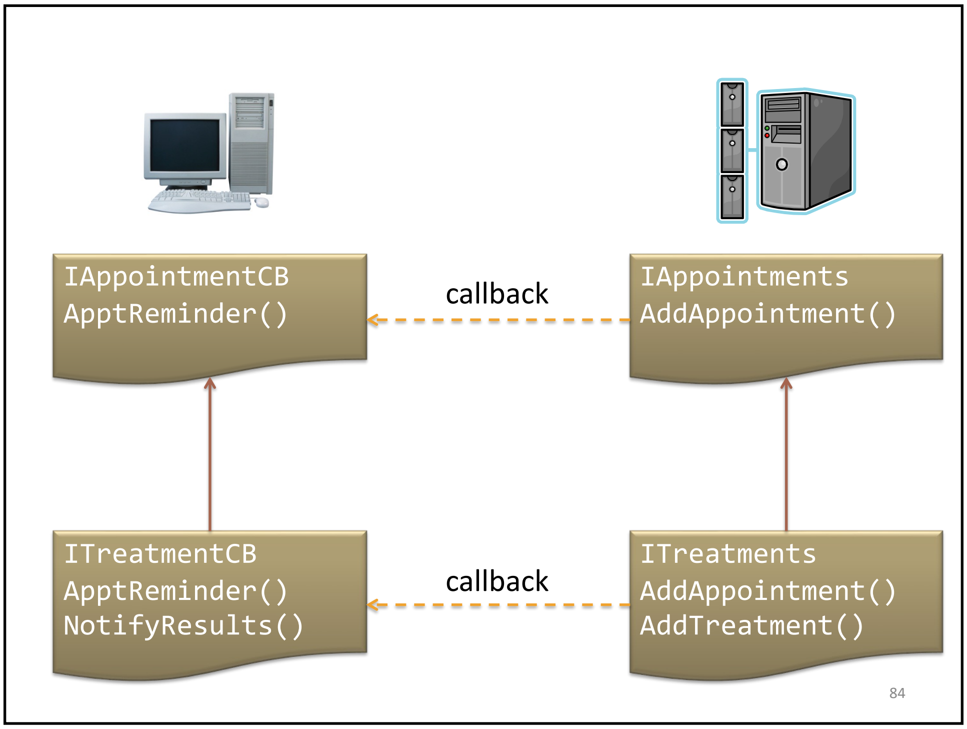
81

COMPATIBLE CHANGE: OPERATIONS CONTRACT



Callback Contract Hierarchy (WCF)

```
[ServiceContract(CallbackContract=
                 typeof(IAppointmentCB))]
interface IAppointments {
   [OperationContract]
   void AddAppointment(...);
}

interface IAppointmentCB {
   [OperationContract]
   void ApptReminder(...);
}
```



IAppointmentCB
ApptReminder()

callback

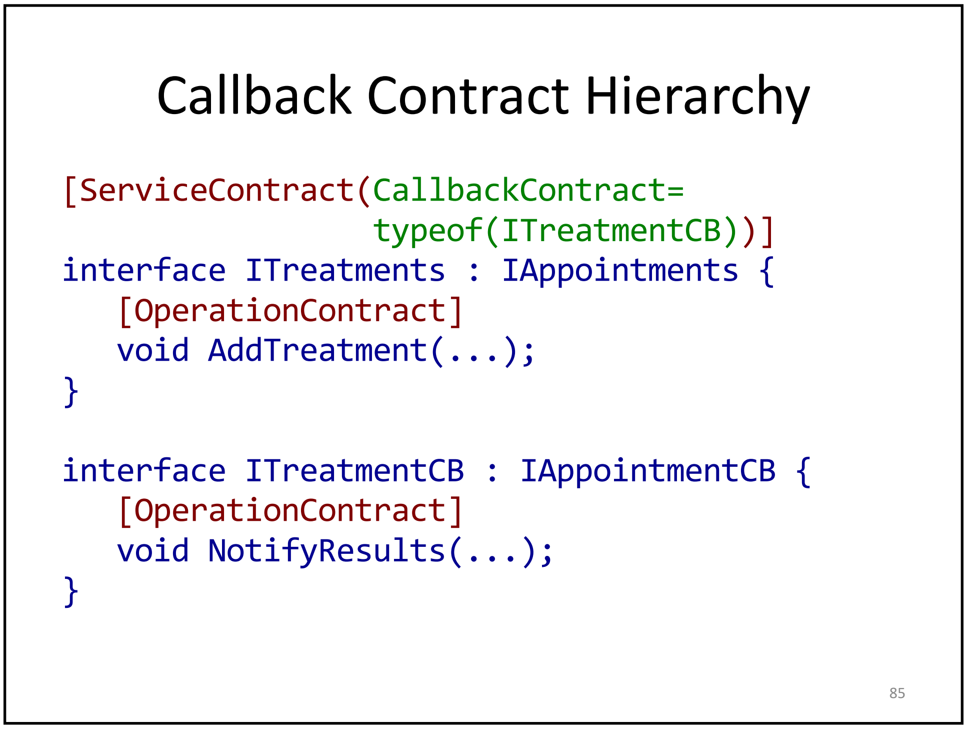
IAppointments
AddAppointment()

ITreatmentCB
ApptReminder()
NotifyResults()

callback

ITreatments
AddAppointment()
AddTreatment()

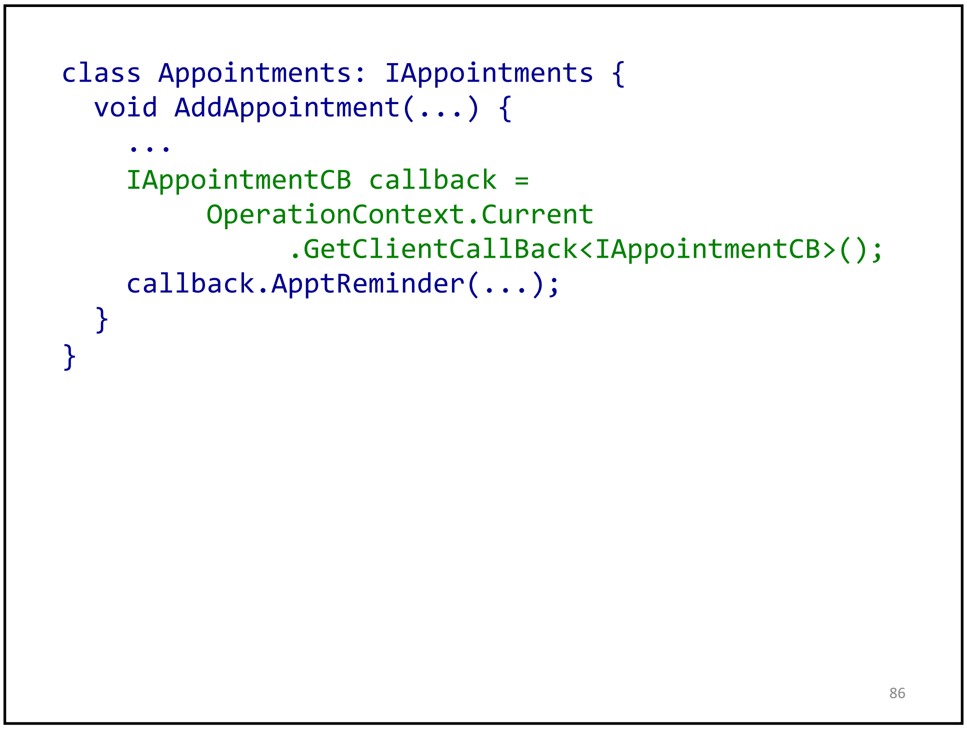
84



Callback Contract Hierarchy

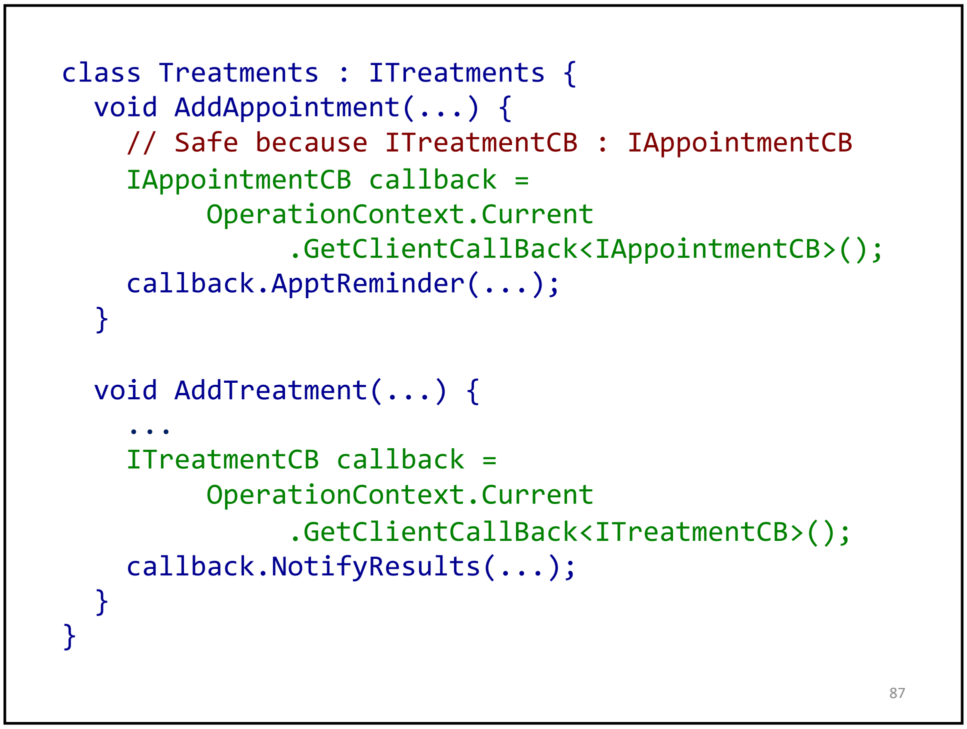
```
[ServiceContract(CallbackContract=
                 typeof(ITreatmentCB))]
interface ITreatments : IAppointments {
   [OperationContract]
   void AddTreatment(...);
}

interface ITreatmentCB : IAppointmentCB {
   [OperationContract]
   void NotifyResults(...);
}
```

85



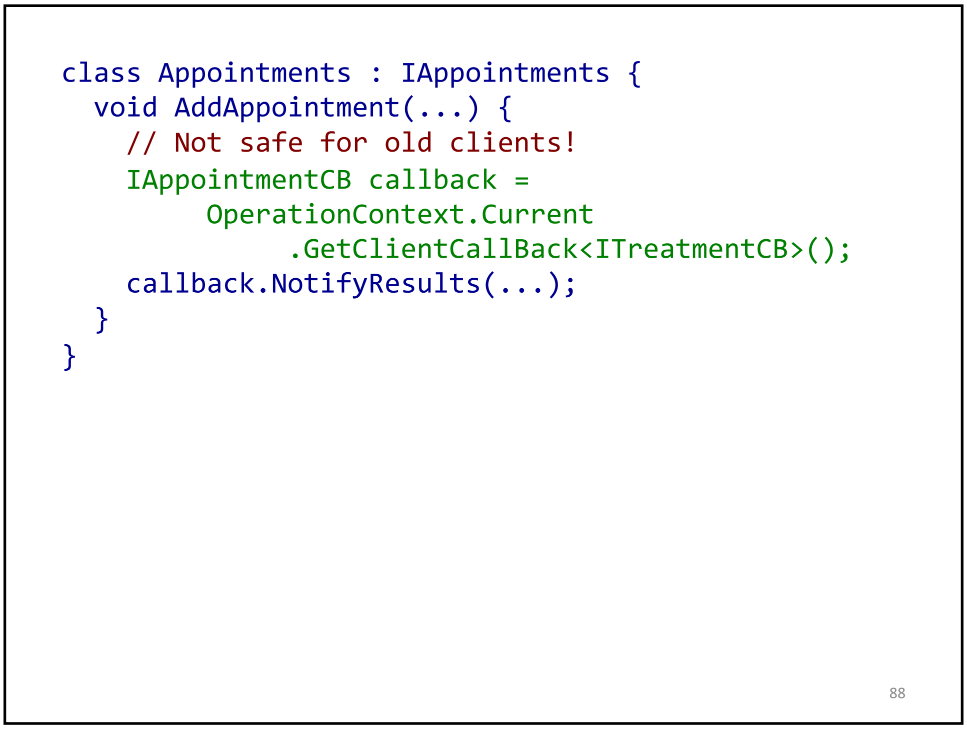
```
class Appointments: IAppointments {
  void AddAppointment(...) {
    ...
    IAppointmentCB callback =
         OperationContext.Current
              .GetClientCallBack<IAppointmentCB>();
    callback.ApptReminder(...);
  }
}
```

86

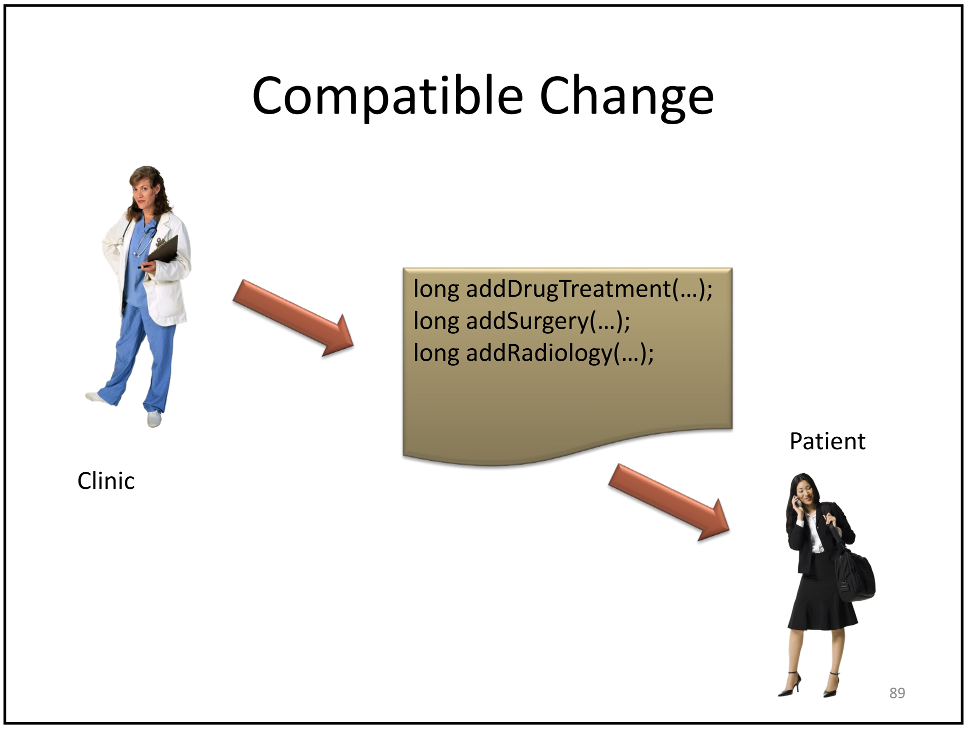


```
class Treatments : ITreatments {
  void AddAppointment(...) {
    // Safe because ITreatmentCB : IAppointmentCB
    IAppointmentCB callback =
         OperationContext.Current
              .GetClientCallBack<IAppointmentCB>();
    callback.ApptReminder(...);
  }

  void AddTreatment(...) {
    ...
    ITreatmentCB callback =
         OperationContext.Current
              .GetClientCallBack<ITreatmentCB>();
    callback.NotifyResults(...);
  }
}
```

87



```
class Appointments : IAppointments {
  void AddAppointment(...) {
    // Not safe for old clients!
    IAppointmentCB callback =
         OperationContext.Current
              .GetClientCallBack<ITreatmentCB>();
    callback.NotifyResults(...);
  }
}
```

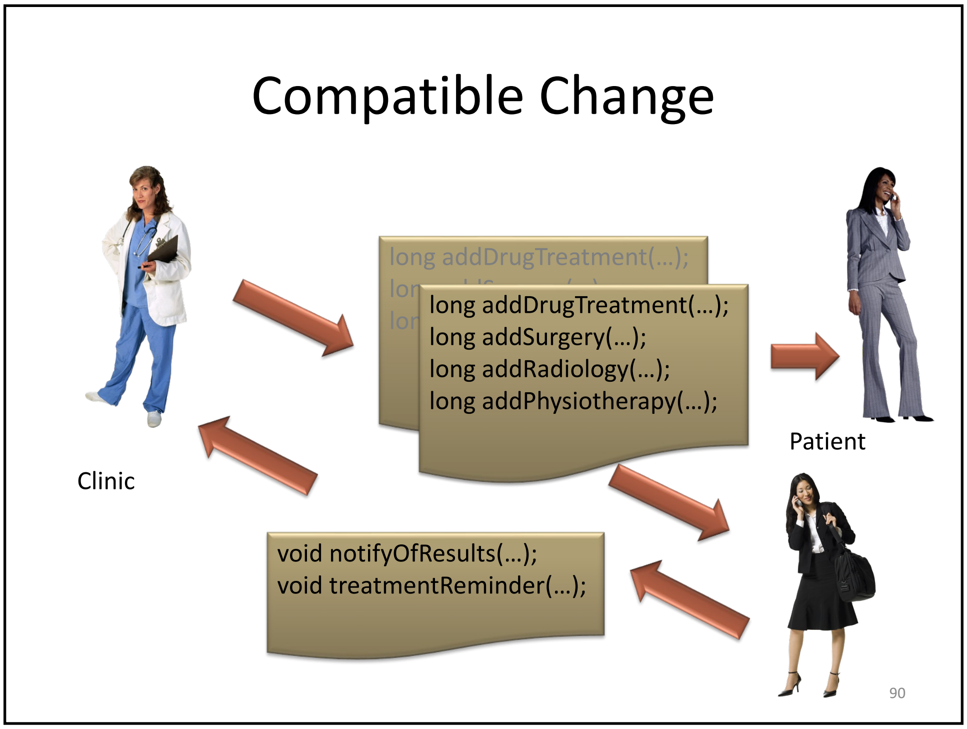


Compatible Change

long addDrugTreatment(…);
long addSurgery(…);
long addRadiology(…);

Clinic

Patient



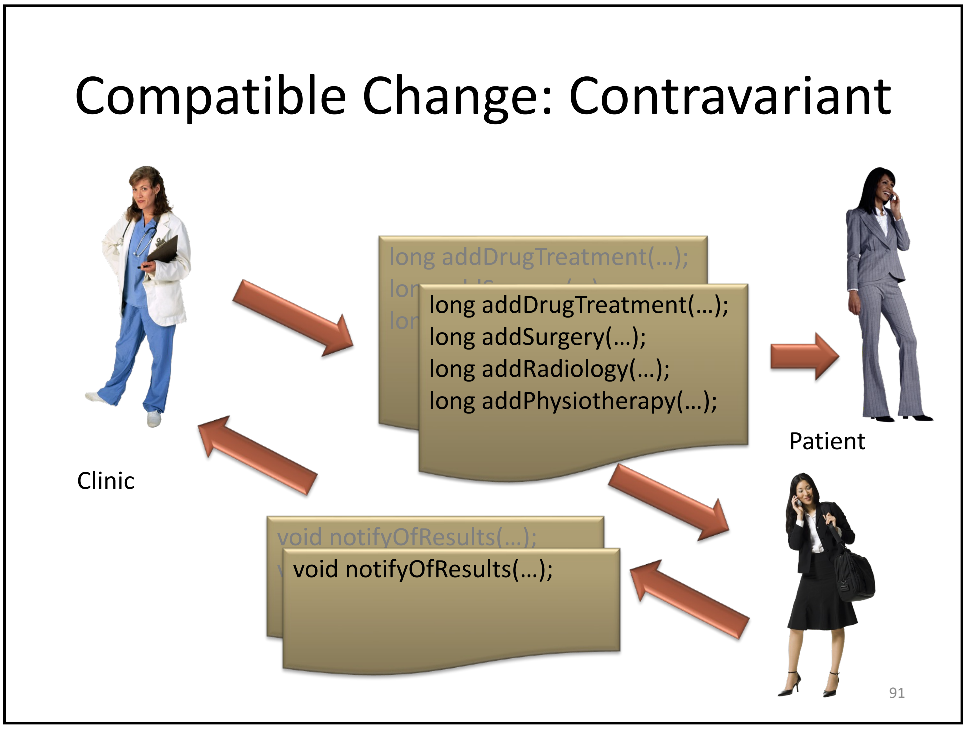
Compatible Change

long addDrugTreatment(…);
lon
lor

long addDrugTreatment(…);
long addSurgery(…);
long addRadiology(…);
long addPhysiotherapy(…);

void notifyOfResults(…);
void treatmentReminder(…);

90



Compatible Change: Contravariant

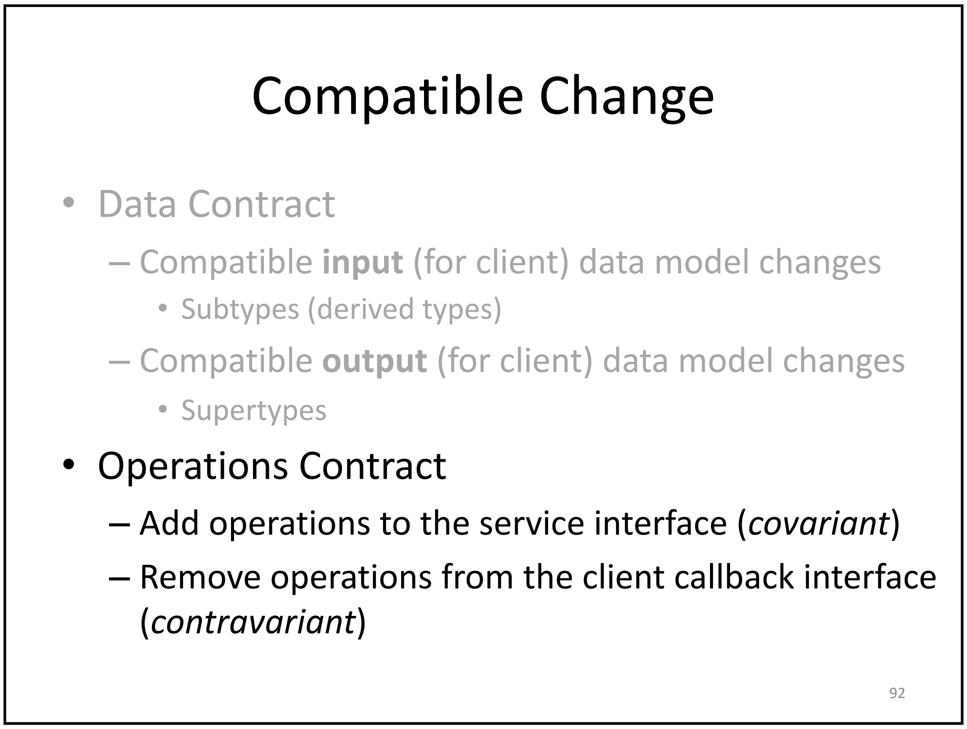
long addDrugTreatment(…);
lon
lor

long addDrugTreatment(…);
long addSurgery(…);
long addRadiology(…);
long addPhysiotherapy(…);

void notifyOfResults(…);

void notifyOfResults(…);

91



Compatible Change

- Data Contract
  - Compatible **input** (for client) data model changes
    - Subtypes (derived types)
  - Compatible **output** (for client) data model changes
    - Supertypes
- Operations Contract
  - Add operations to the service interface (*covariant*)
  - Remove operations from the client callback interface (*contravariant*)