

# Faults and Exceptions

Miguel A. Castro  
@miguelcastro67



# SOAP Faults

- CLR exceptions will not cross machine boundaries
- WCF uses “Soap Faults”
- Specification defining fault message that is packaged and passed from service to client
- WCF provides a .NET programming model to accommodate this without knowing too much about underneath
  - Service throws exception
  - WCF packages it as SOAP fault, embeds it in response message
  - Client recreates received SOAP fault as CLR exception and throws it
  - Client proxy code can be wrapped in standard Try-Catch

# Service Exception Handling Scenarios

- **Unhandled (IncludeExceptionDetailsInFault = false)**
  - Client receives **FaultException**
  - Cannot receive any additional information
  - Proxy faulted
- **Unhandled (IncludeExceptionDetailsInFault = true)**
  - Client receives **FaultException<ExceptionDetail>**
  - Can read exception message
  - Can read original exception type
  - Proxy faulted
  - Warning: this Service Behavior can be altered by admin

# Service Exception Handling Scenarios

- **FaultException** thrown
  - Client receives **FaultException**
  - Can read exception message
  - Proxy OK
  - Note: client can only catch one exception type
- **FaultException<T>** thrown
  - Client receives **FaultException<T>**
  - Can read exception message
  - Proxy OK
  - Note: client can catch specific exceptions
  - **T** can be exception or custom fault contract
  - **T** must be known ahead of time and service contract tagged appropriately
    - **FaultContract** attribute

**Demo Time**