

Ikarus::KirchhoffLoveShell  
::calculateVectorImpl

Ikarus::LinearElastic  
::calculateVectorImpl

Ikarus::NonLinearElastic  
::calculateVectorImpl

Ikarus::Traction::calculate  
VectorImpl

```
graph LR; A[Ikarus::KirchhoffLoveShell::calculateVectorImpl] --> D[Ikarus::Traction::calculateVectorImpl]; B[Ikarus::LinearElastic::calculateVectorImpl] --> D; C[Ikarus::NonLinearElastic::calculateVectorImpl] --> D;
```

The diagram illustrates a design pattern where three different classes (KirchhoffLoveShell, LinearElastic, and NonLinearElastic) implement a common interface or base class (Traction). Each class has its own implementation of the calculateVectorImpl method, which is then used by the Traction class to calculate the vector. The Traction class is highlighted in gray, indicating it is the central or target component.