N|C
NCOMP

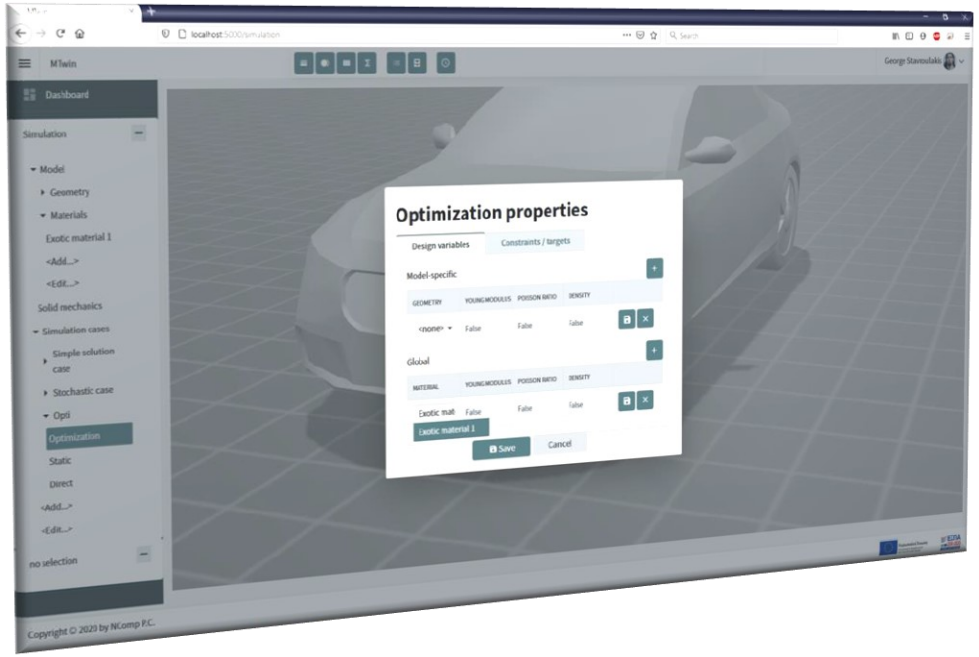
MTWIN
DIGITAL TWINS

Democratizing digital twins technology

EURO
Greece

Συνοπτική παρουσίαση

Ένα καινοτόμο **λογισμικό** *νέφους* με διεπαφή χρήστη *ιστου* για το σχεδιασμό και την κατασκευή ψηφιακών **ομοιοτύπων** για μια ευρεία γκάμα βιομηχανικών τεχνουργημάτων και προϊόντων

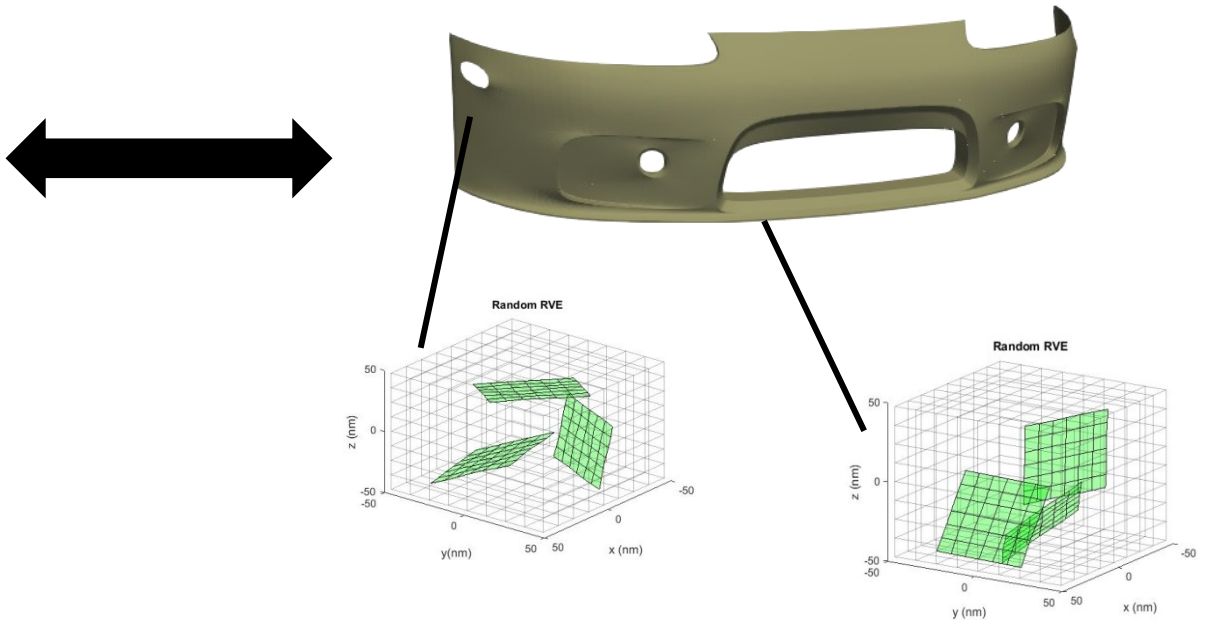


MTwin is the Greek *ABAQUS*, running in your *browser,* on the *cloud*

SIMULIA
ABAQUS

Συνοπτική παρουσίαση

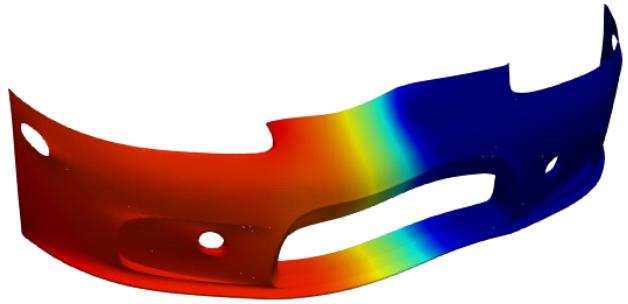
Ψηφιακό ομοιότυπο: ένα **ψηφιακό μοντέλο** ενός τεχνουργήματος που *αναπαριστά τη συμπεριφορά* του τελευταίου



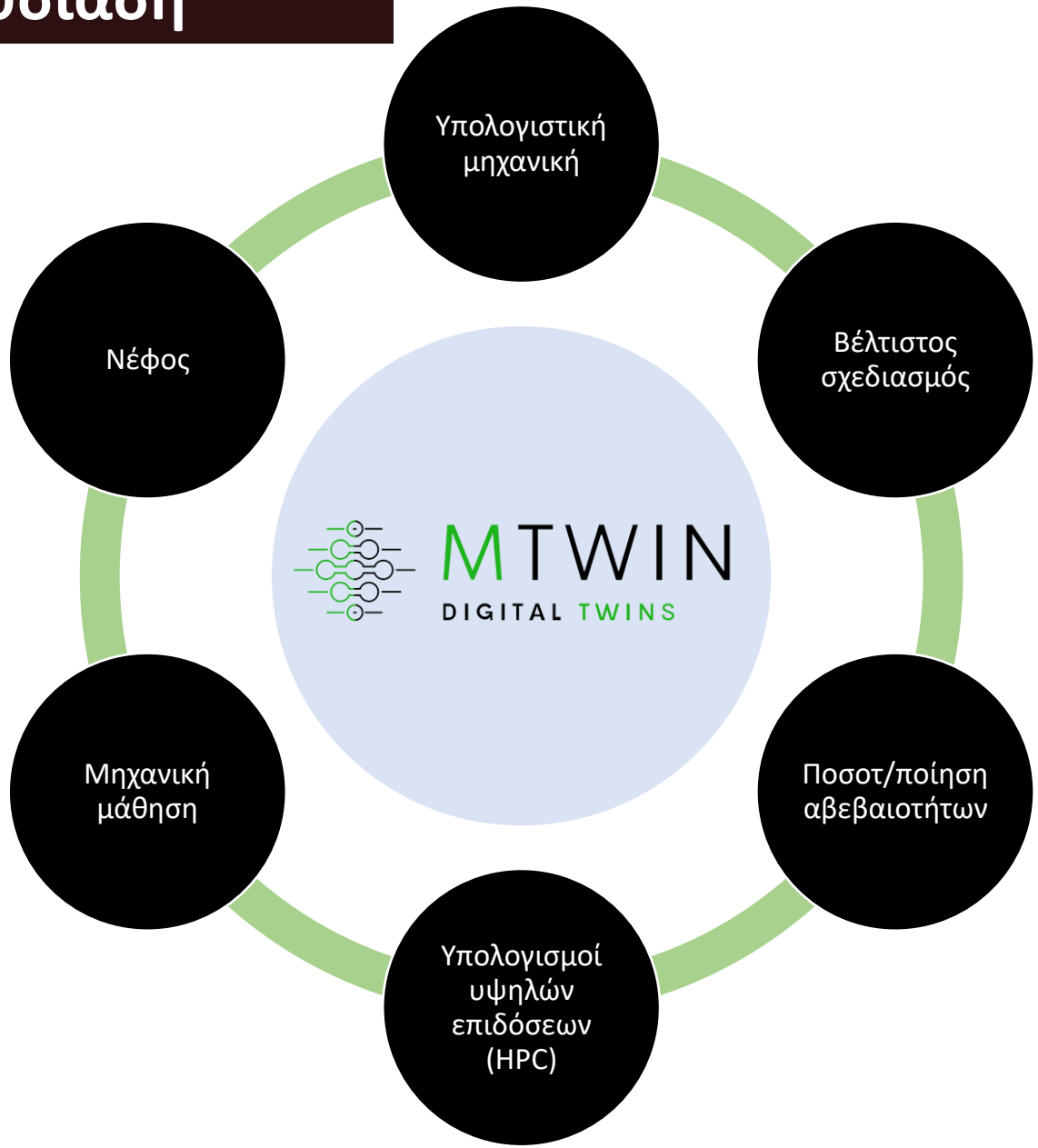
Το **ψηφιακό μοντέλο** υπόκειται στις

Προσδοκούμενες συνθήκες

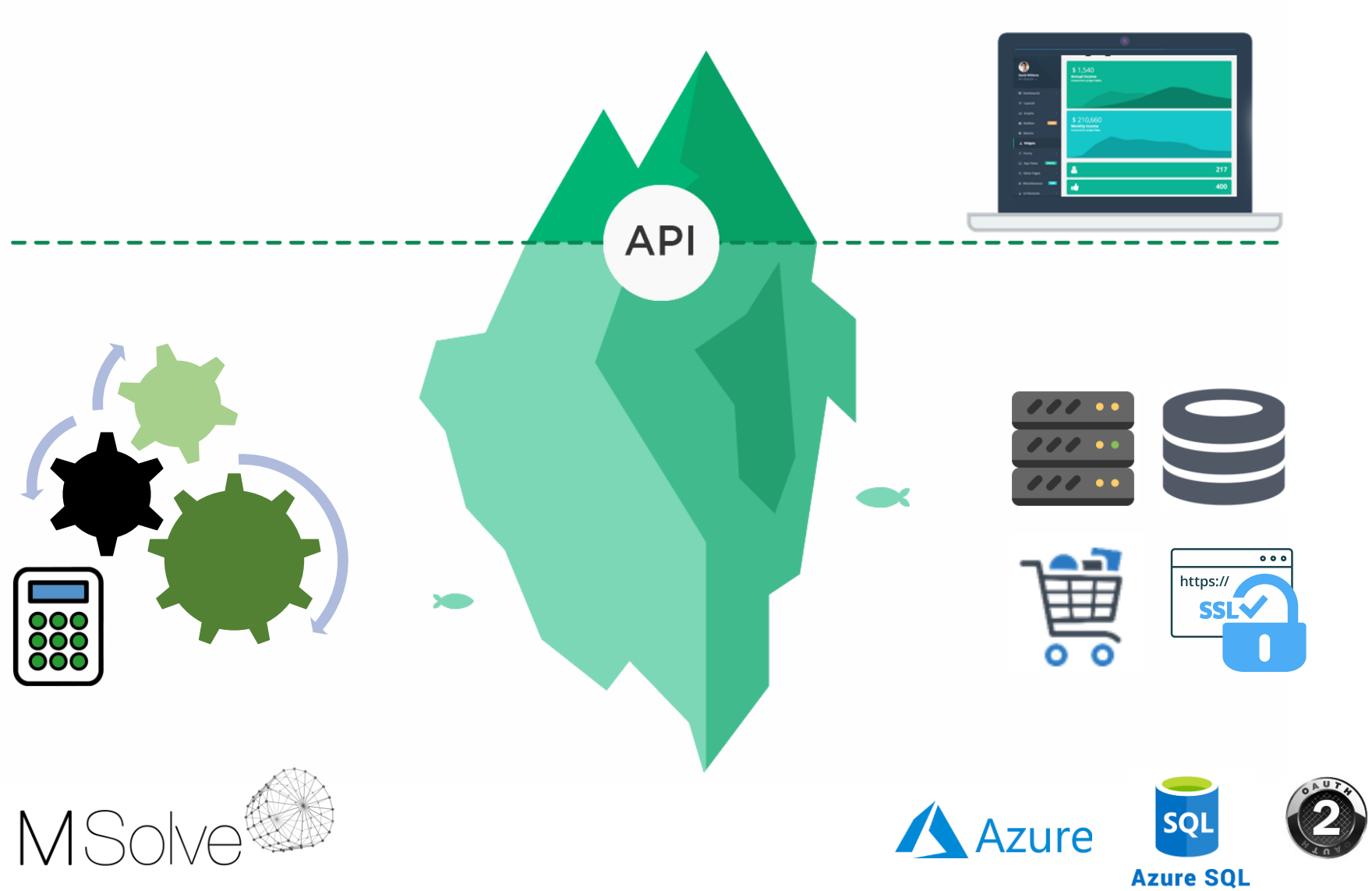
Πραγματικές συνθήκες (*data driven design*)



Συνοπτική παρουσίαση

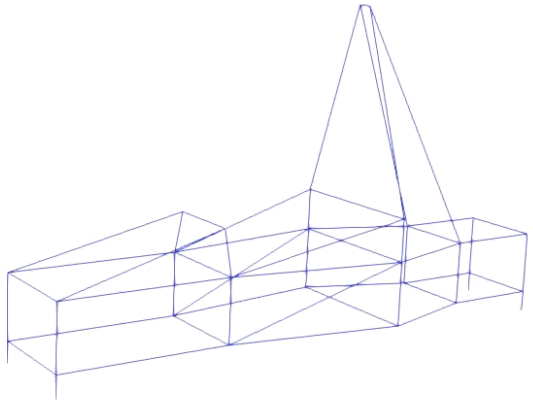


Συνοπτική παρουσίαση

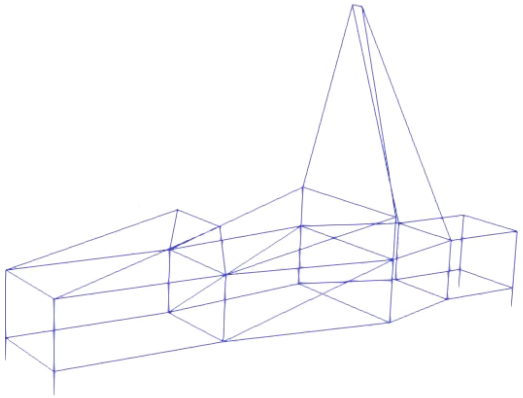


Σενάριο χρήσης HPC

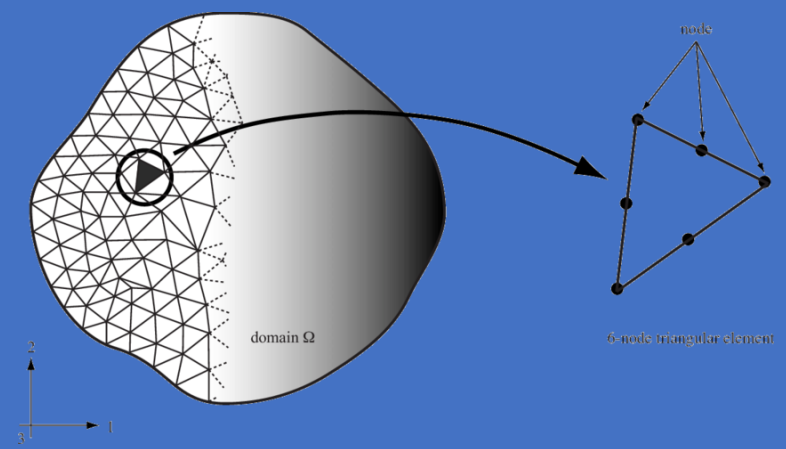
Εταιρεία υποστήριξης αγωνων μηχανοκίνητου αθλητισμου



- Βελτιστοποίηση πλαισίου μονοθεσίου με δεδομένα επιταχυνσιομέτρων
- Βαθμονόμηση παραμέτρων πλαισίου για έλεγχο σε κόπωση και επισκευή



Αρχιτεκτονική MSolve



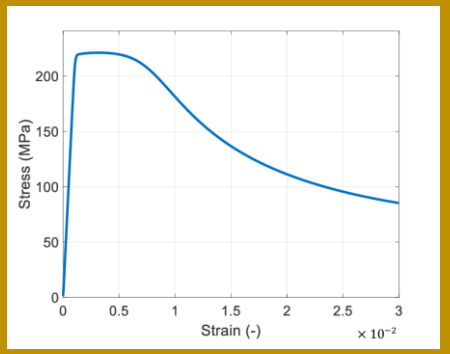
$$N_1 = \frac{1}{4}(1-\xi)(1-\eta)$$

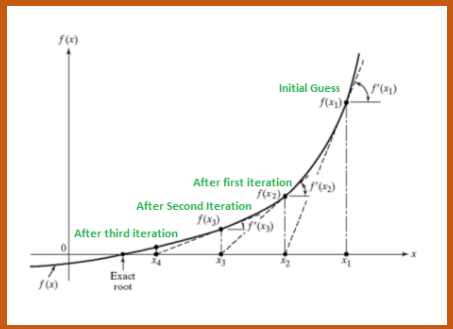
$$N_2 = \frac{1}{4}(1+\xi)(1-\eta)$$

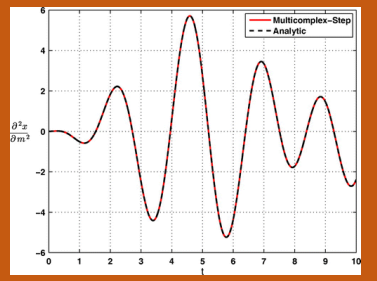
$$N_3 = \frac{1}{4}(1+\xi)(1+\eta)$$

$$N_4 = \frac{1}{4}(1-\xi)(1+\eta)$$

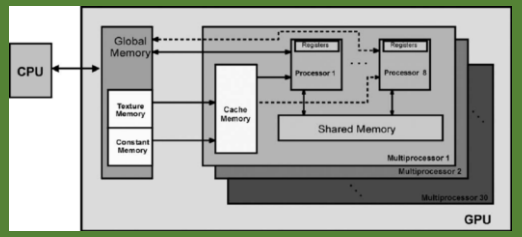
$$\nabla \cdot \boldsymbol{C\varepsilon} = \mathrm{F}$$



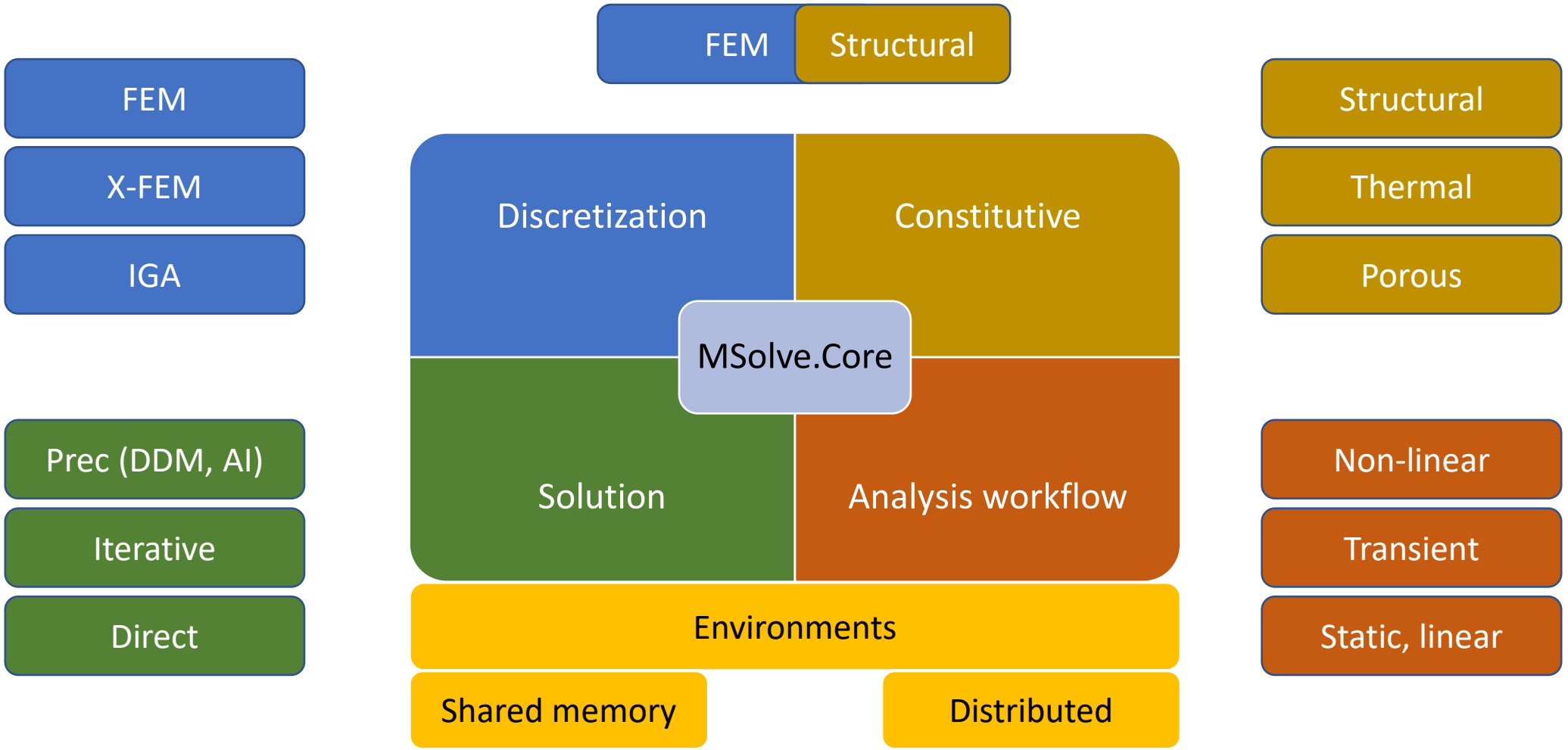




$$\begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix} = \begin{pmatrix} sum_1 \\ sum_2 \\ sum_3 \\ sum_4 \\ sum_5 \end{pmatrix}$$

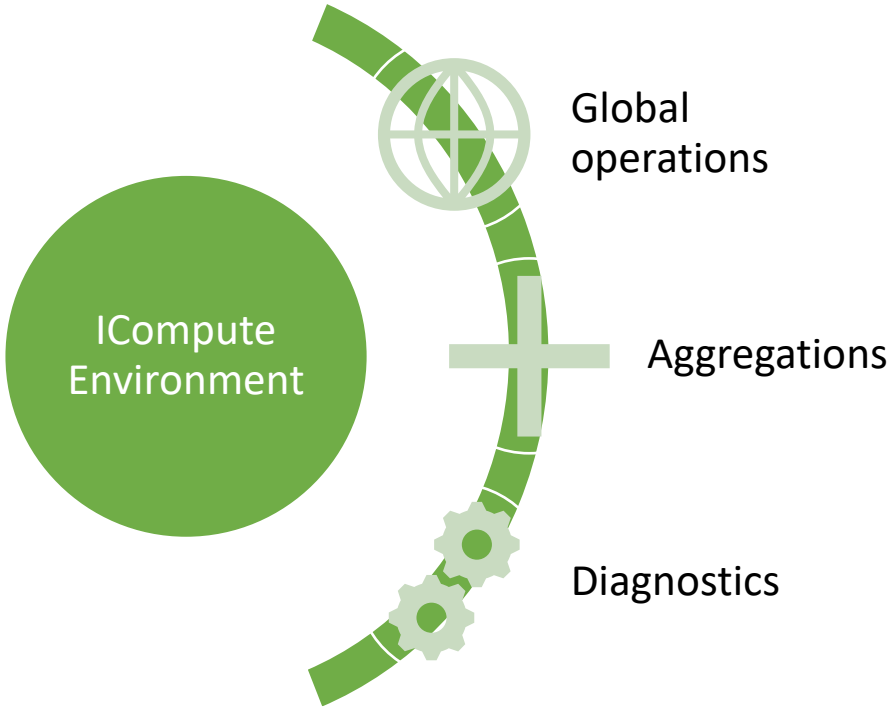


Αρχιτεκτονική MSolve



Ολοκλήρωση HPC MSolve

Υποδομή Environments: Σύνολο *διεπαφών* και *κλάσεων* για την **αφαίρεση παράλληλων** λειτουργιών για **ετερογενή προγραμματιστικά μοντέλα** και **υλικό**




```
12 references
public double Norm2()
{
    Func<int, double> calcLocalDot = node =>
    {
        Vector localVector = this.LocalVectors[node];
        double[] inverseMultiplicities = Indexer.GetLocalComponent(node).InverseMultiplicities;

        double dotLocal = 0.0;
        for (int i = 0; i < localVector.Length; ++i)
        {
            dotLocal += localVector[i] * localVector[i] * inverseMultiplicities[i];
        }

        return dotLocal;
    };

    Dictionary<int, double> dotPerNode = Environment.CalcNodeData(calcLocalDot);
    return Math.Sqrt(Environment.AllReduceSum(dotPerNode));
}
```

Ολοκλήρωση HPC MSolve

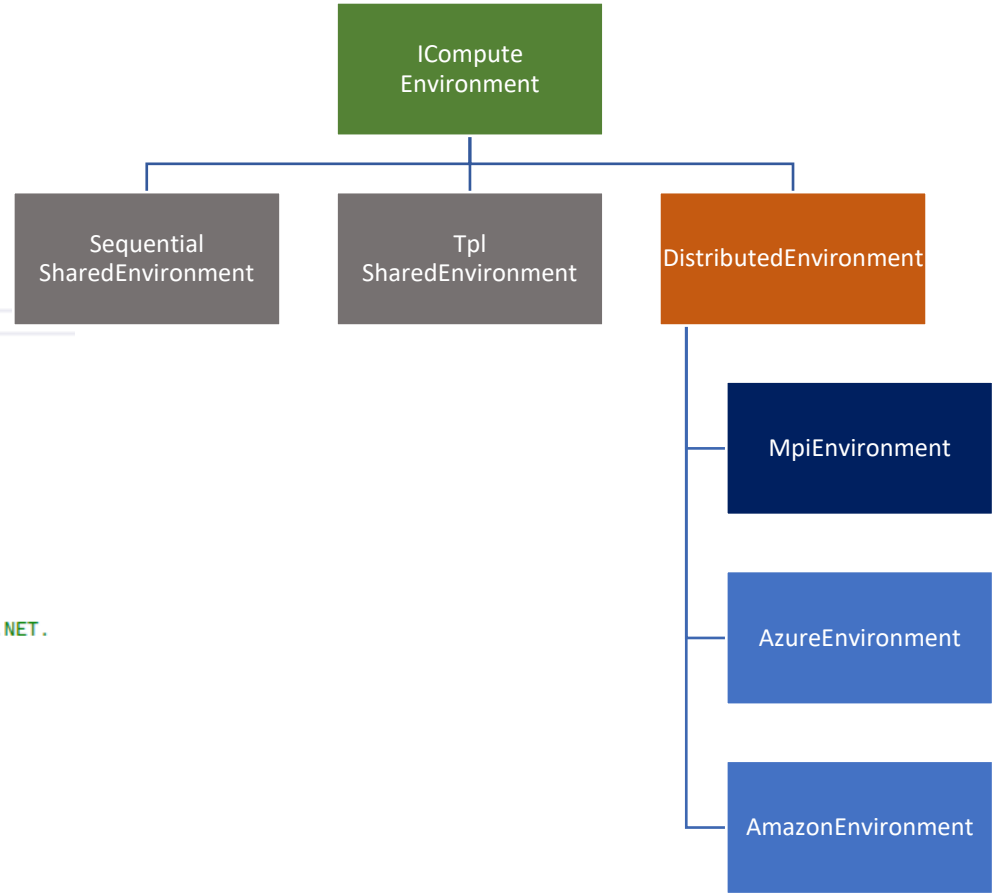
Οι κλάσσεις μοντελοποιούν μια σειρά από υπολογιστικά περιβάλλοντα, μεταξύ των οποίων και τα `MpiEnvironment` για τη διαλειτουργικότητα με το MPI

```
public sealed class AzureEnvironment : IComputeEnvironment, IDisposable
{
    private readonly IDistributedGlobalOperationStrategy globalOperationStrategy;

    private bool disposed = false;
    private Dictionary<int, ComputeNode> localNodes;
    private MpiP2PTransfers p2pTransfers;
    private MpiCollectivesHelper collectivesHelperWorld;
    private MpiCollectivesHelper collectivesHelperNodes;

    0 references
    public AzureEnvironment(IDistributedGlobalOperationStrategy globalOperationStrategy)
    {
        this.globalOperationStrategy = globalOperationStrategy;

        //TODOMPI: See Threading param. In multithreaded programs, I must specify that to MPI.NET.
        string[] args = Array.Empty<string>();
```

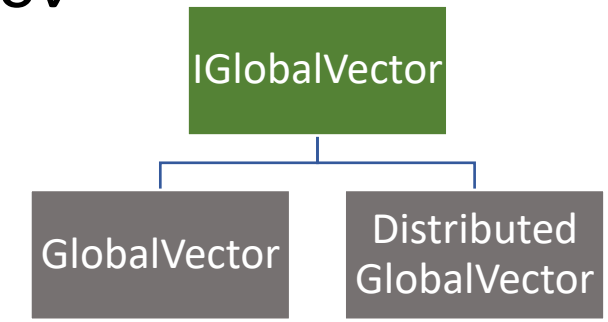


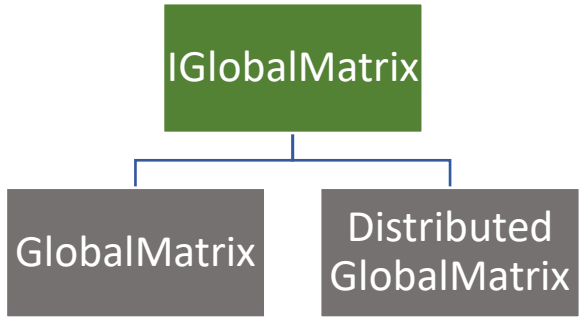
Ολοκλήρωση HPC MSolve

Ενα σύνολο *θεμελιωδών* δομών δεδομένων γραμμικής άλγεβρας εχει υλοποιηθεί με τη χρήση της διεπαφής `IComputeEnvironment`, παρέχοντας μια εμπειρία προγραμματισμού όπως σε ένα σειριακό περιβάλλον

```
10 references
public void LinearCombinationIntoThis(
    double thisCoefficient, IGlobalVector otherVector, double otherCoefficient)
{
    DistributedOverlappingVector otherDistributed = Indexer.CheckCompatibleVector(otherVector);
    LinearCombinationIntoThis(thisCoefficient, otherDistributed, otherCoefficient);
}

2 references
public void LinearCombinationIntoThis(
    double thisCoefficient, DistributedOverlappingVector otherVector, double otherCoefficient)
{
    Indexer.CheckCompatibleVector(otherVector);
    Environment.DoPerNode(
        node => this.LocalVectors[node].LinearCombinationIntoThis(
            thisCoefficient, otherVector.LocalVectors[node], otherCoefficient)
    );
}
```






```
        // The default Fletcher-Reeves formula is: β = δnew / δold = (sNew * rNew) / (sOld * rOld)
        // However we could use a different one, e.g. for variable preconditioning Polak-Ribiere is usually better.
        paramBeta = betaCalculation.CalculateBeta(this);

        // d = s + β * d
        //This allocates a new vector d, copies r and GCs the existing d.
        direction.LinearCombinationIntoThis(paramBeta, precondResidual, 1.0); //This performs additions instead of copying and needless multiplications.
```

N C O M P

MTWIN

DIGITAL TWINS

Questions and discussion

EURO

Greece