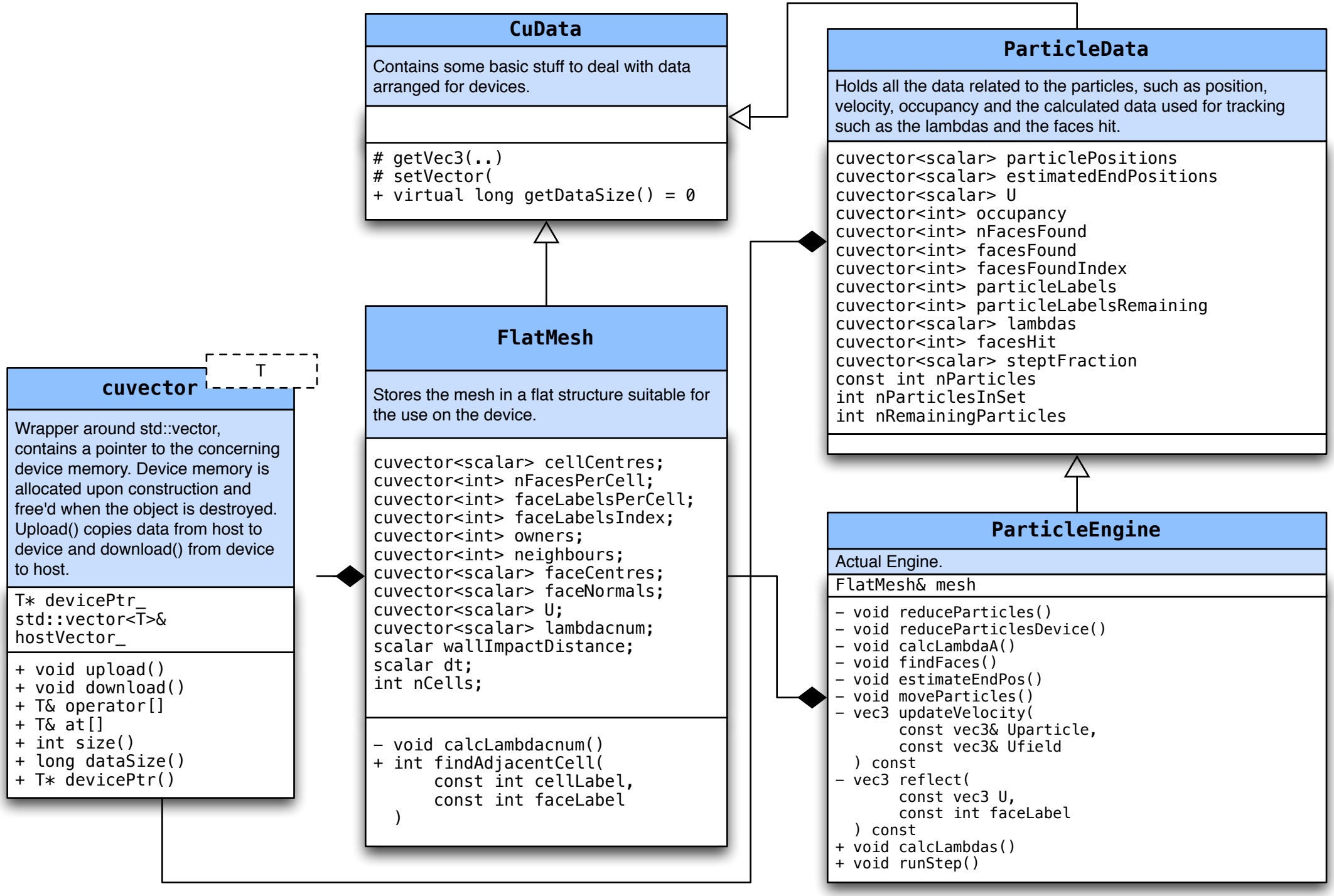
CuData
Contains some basic stuff to deal with data arranged for devices.
getVec3(..)
setVector(
+ virtual long getDataSize() = 0
ParticleData
Holds all the data related to the particles, such as position, velocity, occupancy and the calculated data used for tracking such as the lambdas and the faces hit.
cuvector<scalar> particlePositions
cuvector<scalar> estimatedEndPositions
cuvector<scalar> U
cuvector<int> occupancy
cuvector<int> nFacesFound
cuvector<int> facesFound
cuvector<int> facesFoundIndex
cuvector<int> particleLabels
cuvector<int> particleLabelsRemaining
cuvector<scalar> lambdas
cuvector<int> facesHit
cuvector<scalar> steptFraction
const int nParticles
int nParticlesInSet
int nRemainingParticles
FlatMesh
Stores the mesh in a flat structure suitable for the use on the device.
cuvector<scalar> cellCentres;
cuvector<int> nFacesPerCell;
cuvector<int> faceLabelsPerCell;
cuvector<int> faceLabelsIndex;
cuvector<int> owners;
cuvector<int> neighbours;
cuvector<scalar> faceCentres;
cuvector<scalar> faceNormals;
cuvector<scalar> U;
cuvector<scalar> lambdacnum;
scalar wallImpactDistance;
scalar dt;
int nCells;
- void calcLambdacnum()
+ int findAdjacentCell(
const int cellLabel,
const int faceLabel
)
T
cuvector
Wrapper around std::vector, contains a pointer to the concerning device memory. Device memory is allocated upon construction and free'd when the object is destroyed. Upload() copies data from host to device and download() from device to host.
T* devicePtr_
std::vector<T>& hostVector_
+ void upload()
+ void download()
+ T& operator[]
+ T& at[]
+ int size()
+ long dataSize()
+ T* devicePtr()
ParticleEngine
Actual Engine.
FlatMesh& mesh
- void reduceParticles()
- void reduceParticlesDevice()
- void calcLambdaA()
- void findFaces()
- void estimateEndPos()
- void moveParticles()
- vec3 updateVelocity(
const vec3& Uparticle,
const vec3& Ufield
) const
- vec3 reflect(
const vec3 U,
const int faceLabel
) const
+ void calcLambdas()
+ void runStep()