Current name	New name	3 let	Meaning
	BodyState	Bst	*information about a body that changes during the simulation process* and *is different for each instance of a body in the simulation, like position, velocity, acceleration* and *mass, inertia*
	BodyStateConstraints	Bsc	*information about constraints imposed on a body state, like disallowing to change body's position in z direction, or exceeding a velocity or y position a given maximum limit, a constrained value can be either kept at limiting value, or could decide that body must be deleted (like maximum strain, or body ran out of the simulation edges). This information is respected by engines when the BodyExternalVariables are used to calculate a BodyState for the next execution of simulation loop. Usually a big number of bodies will use the same constraints, and even bigger will not use constraints at all, thus the count of instances of this class will be much smaller than total count of bodies.*
PhysicalParameters	BodyConstitutiveParameters	Bcp	*information about a body that usually does not change during the simulation and is the same for many instances of bodies. For example whole simulation of reinforced concrete will need only two instances of this class: one for concrete, second for reinforcement. It's intended to be an information used by constitutive laws, like stiffness or cohesion*
GeometricalModel	BodyShape	Bsh	*The idealized shape of a body that we want to simulate: it is used to create a simplified shape, and for display*
InteractingGeometry	BodySimplifiedShape	Bss	*a shape of the body used for performing the actual simulation, may be different from idealized shape, because it's merely its representation used for the prupose of simulation*
BoundingVolume	BodyBoundingVolume	Bbv	*a bounding volume is used to detect potential interaction between bodies, usually is built from information stored inside simplified shape*
InteractionGeometry	InteractionState	Ist	*information about an interaction happening between bodies which changes while the interaction evolves during the simulation, like penetration depth or shearing force*
InteractionPhysics	InteractionConstitutiveParameters	Icp	*information about an interaction happening between bodies which usually does not change during the simulation, even when bodies disconnect and reconnect again. Like contact stiffness*
	InteractionShape	Ish	*if an interaction can have an idealized shape representation, this class is for it, like a water meniscus between two spheres in cohesive interaction caused by water.*
	OutputData[1]	Odt	*If you need some information which cannot be directly read from the classes mentioned above, but you need to calculate something and store it somewhere for further reading, use this class. The engines of simulation loop usually will not read from this class, it is rather to store some (final) results needed from the simulation*
PhysicalAction	BodyExternalVariables	Bex	*It is a complementary information about a body, which usually sums the effect of all physical actions on a body. This information most often is an intermediate stage that is used to calculate future values (for the next execution of simulation loop) of data stored in BodyState. For example a sum of forces and moments acting on a sphere (they will be used to change body's position and orientation which are stored in BodyState)*
Engine	Engine	Egi	*Anything that is reading or writing into the simulation data while performing some computational operations, it is a basic building block of the simulation loop. The engine is not supposed to store any information related to calculated model, only information that configures the way in which the engine runs.* See note at *EngineFunctor*
MetaEngine	EngineDispatcher	Egd	
EngineUnit	EngineFunctor	Egf	*Both Engine and Enginefunctor will receive information about current iteration, so they could perform something during only a single iteration on all bodies (eg. on the first call to prepare them for further calculations). It must be done because the loop has to be* ***outside*** *- to allow parallelization of the loop*
DeuxExMachina	EgiConditionApplier	Econ	
InteractionSolver	EgiConstitutiveLaw	Elaw	
BroadInteractor	EgiBoundingVolumeCollider	Ebvc	
DataRecorder	EgiDataProcessor	Edat	
MetaBody	Model	[2]	*This class contains everything inside: the data classes for bodies, interactions, and also a simulation loop with engines inside*
Body			**class is removed, to allow reorganization of containers**

The 3 letter name: is used to **prefix each** class name in yade, to indicate the ROOT of inheritance tree. Yade developers will need to remember those short name versions - thus the intention is to make them easy to remember.

[1] - an idea is to add a class that is used specifically to record and calculate some information about happening simulation (and have an GLDraw handler also, just like all the other classes have it and can be drawn on the screen). It's an intermediate stage between reading directly from the simulation to have results - a stage to process this information a bit ans tore it here.

[2] - you never derive from this class