/\* global getAssetRegistry getFactory emit query \*/

function beginSOWRequest(request){

console.log('begin sow request')

return getAssetRegistry('org.capgemeni.sogeti.SOW')

.then(function(registry){

console.log('got asset registry');

var factory = getFactory();

var newSOW = factory.newResource('org.capgemeni.sogeti', 'SOW', request.sowId)

newSOW.requirement1 = request.requirement1;

newSOW.requirement2 = request.requirement2;

newSOW.requirement3 = request.requirement3;

newSOW.startDate = request.startDate;

newSOW.endDate = request.endDate;

newSOW.status = "Requested"; // enum: requested

newSOW.employee = request.consultant.employee;

newSOW.client = request.client;

return registry.add(newSOW);

});

}

/\*\*

\* Track a client's consultant request

\* @param {org.capgemeni.sogeti.RequestConsultant} request - the request to be processed

\* @transaction

\*/

async function requestConsultant(request) { // eslint-disable-line no-unused-vars

* console.log('in tx');

// (buyer) request.Client

// (asset) request.Consultant

// set the reserving client of the consultant

request.consultant["client"] = request.client;

const assetRegistry = await getAssetRegistry('org.capgemeni.sogeti.Consultant');

// persist the state of the commodity

await assetRegistry.update(request.consultant);

// begin SOW asset request

beginSOWRequest(request);

}

/\*\*

\* Track a company's SOW proposal

\* @param {org.capgemeni.sogeti.ProposeSOW} proposal - the proposal to be processed

\* @transaction

\*/

async function proposeSOW(proposal) { // eslint-disable-line no-unused-vars

console.log('in tx');

// (price proposal) proposal.priceAgreement

// (asset) proposal.sow

// issue the SOW proposal by company, which specifices the term & conditions (priceAgreement)

proposal.sow.priceAgreement = proposal.priceAgreement;

proposal.sow.status = "Proposed"; // enum: proposed

const assetRegistry = await getAssetRegistry('org.capgemeni.sogeti.SOW');

// persist the state of the sow

await assetRegistry.update(proposal.sow);

}

/\*\*

\* Track a client's sow agreement

\* @param {org.capgemeni.sogeti.FinalizeAgreement} agreement - the agreement to be processed

\* @transaction

\*/

async function FinalizeAgreement(agreement) { // eslint-disable-line no-unused-vars

console.log('in tx');

// (terms acceptance) agreement.acceptTerms

// (asset) agreement.sow

// finalize SOW process with client acceptance/denial of terms & conditions (priceAgreement)

agreement.sow.status = agreement.acceptTerms ? "Accepted" : "Rejected";

const assetRegistry = await getAssetRegistry('org.capgemeni.sogeti.SOW');

// persist the state of the sow

await assetRegistry.update(agreement.sow);

}