Relation to the old project

This overview is based on commit `dc96d7f1ec6fd1ca87562e543319a50a541fb144` from June 18th, 2020. It was the most recent one in early August 2020.

Relevant modules and their API access (i.e. the access to functions in `mysqlConnector.js`)

- `storylineBAforce` (or one of the derived modules) calls: `getStoryline`, `getStoryline_depth1`, `getStoryline_depthN`
- `sl_kontaktnetzwerk` (that is the one with forward- & backward tracking) calls: `getStoryline`, `getStoryline_depth1`, `getStoryline_depthN`
- `linelist` calls: `getTimeline`, `getTimeline_depth1`, `getTimeline_depthN`
- `epedemiekurve` calls: `getEpidemiKeime`, `getEpidemiKeimeCS`

Called stored procedures

These are the stored procedures of the SQL data base, and the API functions in `mysqlConnector.js` that use them. The list is sorted in the order of appearance in the javascript source code:

- `Contact_Station_EPs` by: getUncertaintyGraph, recursiveUncertaintyGraph
- `Patient_Bewegung_Ps` by: getStoryline, getTimeline, getTimeline_depth1, getTimeline_depthN, getStoryline_depth1, getStoryline_depthN
- `Patient_MikroDaten_Ps` by: getStoryline, getTimeline, getTimeline_depth1, getTimeline_depthN, getStoryline_depth1, getStoryline_depthN
- `GetHospitals` by: GetHospitals
- `GetStationBezByStationList` by: GetStationBezByStationList, getStorylineKontaktnetz, getEpidemiStationen, getBarChartStation, getKontaktnetz, getKontakte
- `Station_H` by: Station_H
- `Station_Ss` by: Station_Ss
- `GetPatientCountByTimeHospital` by: GetPatientCountByTimeHospital
- `GetPatientsByTimeHospital` by: GetPatientsByTimeHospital
- `GetPatientInfo` by: GetPatientInfo
- `GetPatientInfoByPatientList` by: GetPatientInfoByPatientList
- `GetErsterkrankungByPatientKeim` by: GetErsterkrankungByPatientKeim
- `GetErsterkrankungenByPatient` by: GetErsterkrankungenByPatient
- `GetPatientsInfoByTimeKeimStation` by: GetPatientsInfoByTimeKeimStation, getBarChartStation
- `IsPatientKrankByKeimPatientList` by: IsPatientKrankByKeimPatientList, getStorylineKontaktnetz, getStorylineINFpatientsT1grade, getStorylineINFpatientsT, getStorylineINFpatientsTbisT1grade, getKontaktnetz, getKontakte
- `GetKeime` by: GetKeime
- `GetKeimBezByKeimList` by: GetKeimBezByKeimList, getEpidemiKeime, getEpidemiKeimeCS
- `GetPositiveMikroDataByTime` by: GetPositiveMikroDataByTime
- `GetDistinctPositiveMikroDataByTime` by: GetDistinctPositiveMikroDataByTime
- `GetPositiveMikroDataByTimeKeim` by: GetPositiveMikroDataByTimeKeim

- `GetDistinctPositiveMikroDataByTimeKeim` by: GetDistinctPositiveMikroDataByTimeKeim
- `GetMikroDataByTimeKeim` by: GetMikroDataByTimeKeim
- `GetMikroDataGroupedbyKeimDateByTimeKeimList` by: GetMikroDataGroupedbyKeimDateByTimeKeimList, getEpidemiKeime
- `cs_GetMikroDataGroupedbyKeimDateByTimeKeimList` by: getEpidemiKeimeCS
- `GetMikroDataGroupedbyStationDateByTimeKeim` by: GetMikroDataGroupedbyStationDateByTimeKeim, getEpidemiStationen, GetPatientInfo
- `GetContacts` by: GetContacts
- `GetContactsByHospital` by: GetContactsByHospital
- `GetContactsByTime` by: GetContactsByTime
- `GetContactsByTimeHospital` by: GetContactsByTimeHospital
- `GetContactNetworkByTimePatient` by: GetContactNetworkByTimePatient, getKontaktnetz, getTimeline_depth1, getStoryline_depth1
- `GetContactsNthDegreeByTimePatientDegree` by: GetContactsNthDegreeByTimePatientDegree, getKontakte, getTimeline_depthN, getStoryline_depthN
- `GetAnsteckungspotentialByTimePatientsKeim` by: GetAnsteckungspotentialByTimePatientsKeim
- `GetMaterial` by: GetMaterial
- `GetMaterialBezByMaterialList` by: GetMaterialBezByMaterialList, getBarChartStation
- `Patient_Bewegung_TTHPs` by: Patient_Bewegung_TTHPs
- `Misc_Bewegungsart` by: Misc_Bewegungsart
- `Misc_Bewegungstyp` by: Misc_Bewegungstyp
- `Patient_Bewegung_tHPs_lastBeforeT` by: Patient_Bewegung_tHPs_lastBeforeT, getStorylineKontaktnetz, getStorylineINFpatientsT1grade, getStorylineINFpatientsT, getStorylineINFpatientsTbisT1grade
- `Contact_PatientList_THPs` by: Contact_PatientList_THPs, getStorylineINFpatientsT1grade, getStorylineINFpatientsT
- `Contact_PatientList_TTHPs` by: Contact_PatientList_TTHPs, getStorylineKontaktnetz, getStorylineINFpatientsT1grade, getStorylineINFpatientsT, getStorylineINFpatientsTbisT1grade
- `Patient_TTH` by: Patient_TTH
- `Patient_Ersterkrankung_TTHK` by: Patient_Ersterkrankung_TTHK, getStorylineINFpatientsTbisT1grade
- `Patient_Erkrankt_TTHK` by: Patient_Erkrankt_TTHK, getStorylineINFpatientsT1grade, getStorylineINFpatientsT
- `Labor_ErregerProTag_TTEsK` by: Labor_ErregerProTag_TTEsK
- `Labor_ErregerProTag_TTEsKSs` by: Labor_ErregerProTag_TTEsKSs
- `GetAllMikroDataInformationByTimeKeimListofPatientHospital` by: GetAllMikroDataInformationByTimeKeimListofPatientHospital, getStorylineKontaktnetz, getStorylineINFpatientsT1grade, getStorylineINFpatientsT, getStorylineINFpatientsTbisT1grade, getKontaktnetz, getKontakte

Ignored everything called by these functions:

- `getStoryline_DEMO` (has "DEMO" in name)
- `GetDBInformation` (will probably not be portable at all)

- `getStorylineINFpatientsTbisT` ("nur temporär, um "irgendwas" zu bekommen, da Filter nicht voll funktional ist")
- `getKontakt` (below comment "TEST FUNCTIONS AB HIER")
- `testFunc` (below comment "TEST FUNCTIONS AB HIER")
- `getHypothesisGantt` (below comment "TEST FUNCTIONS AB HIER")

Used calls available in the old mysqlConnector.js

Combining the relevant modules from above and the stored procedures which they call though the API functions, we arrive at the following list of stored procedures which we must support in the new backend. Additionally, the archetypes of the *HiGHmed* project providing the relevant data are given, along with translations from the existing parameter names to the new ones.

- `Patient_Bewegung_Ps` ⟶ `Patientenaufenthalt`
    - `id: int` ⟶ `?`
    - `Beginn: datetime` ⟶ `?`
    - `Ende: datetime` ⟶ `?`
    - `PatientID: int` ⟶ `?`
    - `FallID: int` ⟶ `?`
    - `LfdNr: int` ⟶ `?`
    - `StationID: int` ⟶ `?`
    - `CaseID: int` ⟶ `?`
    - `CaseType l: string` ⟶ `?`
    - `CaseType k: string` ⟶ `?`
    - `BewegungsartID: int` ⟶ `?`
    - `Bewegungsart l: string` ⟶ `?`
    - `Bewegungsart k: string` ⟶ `?`
    - `BewegungstypID: int` ⟶ `?`
    - `Bewegungstyp: string` ⟶ `?`
- `Patient_Labordaten_Ps` (previously called `Patient_MikroDaten_Ps`) ⟶ `Mikrobiologischer Befund`
    - `AntibiogrammID: int` ⟶ `?`
    - `LabordatenID: int` ⟶ `?`
    - `PatientID: int` ⟶ `?`
    - `FallID: int` ⟶ `?`
    - `ResultatID: int` ⟶ `?`
    - `ProbeID: int` ⟶ `?`
    - `Auftragsdatum: datetime` ⟶ `?`
    - `Eingangsdatum: datetime` ⟶ `?`
    - `MaterialID: int` ⟶ `?`
    - `Material_l: string` ⟶ `?`
    - `Material_k: string` ⟶ `?`
    - `MaterialKombiID: string` ⟶ `?`
    - `Befund: bool` ⟶ `?`
    - `Befundkommentar: string` ⟶ `?`
    - `KeimID: int` ⟶ `?`
    - `Keim_l: string` ⟶ `?`

    - `Keim_k: string` ⟶ `?`
    - `AntibiotikumID: int` ⟶ `?`
    - `Antibiotikum_l: string` ⟶ `?`
    - `Antibiotikum_k: string` ⟶ `?`
    - `ErgebnisID: int` ⟶ `?`
    - `Ergebnis_l: string` ⟶ `?`
    - `Ergebnis_k: string` ⟶ `?`
- `Contact_1stDegree_TTPK` (previously called `GetContactNetworkByTimePatient`) ⟶ `?` -> remove, make special case of Nth Degree
    - `paID: int` ⟶ `?`
    - `pbID: int` ⟶ `?`
    - `Beginn: datetime` ⟶ `?`
    - `Ende: datetime` ⟶ `?`
    - `StationID: int` ⟶ `?`
- `Contact_NthDegree_TTKP_Degree` (previously called `GetContactsNthDegreeByTimePatientDegree`) ⟶ like above, but with one additional field; why not merge all the degree X procedures into one with an optional degree parameter and make 1st degree simply a special case of the Nth degree?
    - `Grad: [?]` ⟶ `?`
- `GetKeimBezByKeimList` ⟶ only used in conjunction with `GetMikroDataGroupedbyKeimDateByTimeKeimList`/`cs_GetMikroDataGroupedbyKeimDateByTimeKeimList` and can probably be simply included in them
    - `id: int`
    - `BEZL: string`
    - `BEZK: string`
- `Labor_ErregerProTag_TTEsK` (previously called `GetMikroDataGroupedbyKeimDateByTimeKeimList`) ⟶ `?`
    - `Datum: datetime` ⟶ `?`
    - `ErregerID: int` ⟶ `?`
    - `ErregerBEZL: string` ⟶ `?`
    - `ErregerBEZK: string` ⟶ `?`
    - `Anzahl: int` ⟶ `?`
    - `Anzahl_cs: int` ⟶ `?`
    - `MAVG7_cs: int` ⟶ `?`
    - `MAVG28_cs: int` ⟶ `?`
- `Labor_ErregerProTag_TTEsK_cs` (previously called `cs_GetMikroDataGroupedbyKeimDateByTimeKeimList`) ⟶ like above, only additionally filtered for copy strains (i.e. additional redundant probes taken from the same patient) -> can be left out