```
package org.ekkescorner.dsl.qtsummit.cascades {

    import org.lunifera.dsl.ext.cpp.qt.datatypes.*;
    import org.lunifera.dsl.ext.cpp.qt.lib.types.annotation.*;

    @TargetOS("Qt")
    @FriendsClass("DataUtil,DataServer")
    dto SettingsData {
        domainKey int id;
        var int version;
        var QString apiVersion;
        var Timestamp lastUpdate;
        var bool isProductionEnvironment;
        var int primaryColor;
        var int accentColor;
        var bool darkTheme;
        var bool useMarkerColors;
        var bool defaultMarkerColors;
        var QString markerColors;
        var bool hasPublicCache;
        var bool useCompactJsonFormat;
        var int lastUsedNumber;
        var QString publicRoot4Dev;
        var bool autoUpdate;
        var int autoUpdateEveryHours;
        var Timestamp lastUpdateStamp;
    }

    @CachePolicy("-R-")
    @TargetOS("Qt")
    dto Conference {
        domainKey int id;
        var QString conferenceName;
        var QString conferenceCity;
        var QString address;
        // TODO not used yet
        var QString timeZoneName;
        var int timeZoneOffsetSeconds;
        @DateFormatString("yyyy-MM-dd")
        var Date conferenceFrom;
        @DateFormatString("yyyy-MM-dd")
        var Date conferenceTo;
        var QString hashTag;
        var QString homePage;
        // TODO Map Integration GeoCoordinate
```

```
    var QString coordinate;
    var int lastBuildingId;
    var int lastFloorId;
    var int lastRoomId;
    var int lastSessionTrackId;
    // implicitly ref lazy
    // sorted: day,start
    var Day [ 0 .. * ] days;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto Building {
    domainKey int buildingId;
    var QString buildingName;
    var bool inAssets;
    // implicitly ref lazy
    var Floor [ 0 .. * ] floors;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto Floor {
    domainKey int floorId;
    var QString floorName;
    var bool inAssets;

    ref lazy Building [1] building;
    // implicitly ref lazy
    var Room [ 0 .. * ] rooms;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto Room {
    domainKey int roomId;
    var QString roomName;
    var bool inAssets;

    ref lazy Floor [1] floor;
    // implicitly ref lazy
    // sorted day,start
    var Session [ 0 .. * ] sessions;
}
```

```
@CachePolicy("-R-")
@TargetOS("Qt")
dto Session {
    domainKey int sessionId;
    var bool isDeprecated;
    // sort day start 010900
    var QString sortKey;
    var bool isTraining;
    var bool isLightning;
    var bool isKeynote;
    var bool isSession;
    var bool isCommunity;
    var bool isUnconference;
    var bool isMeeting;
    var QString title;
    var QString subtitle;
    var QString description;
    var QString sessionType;
    @DateFormatString("HH:mm")
    var Time startTime;
    @DateFormatString("HH:mm")
    var Time endTime;
    var int minutes;
    var QString abstractText;

    // don't want to cache updated Sessions
    // so isFavorite and isBookmarked are updated at Startup
    transient bool isFavorite;
    transient bool isBookmarked;

    // implicitly ref lazy
    var Speaker [0 .. *] presenter;
    var SessionLink [0 .. *] sessionLinks;
    ref lazy Day [ 1 ] sessionDay;
    ref lazy Room [ 0 .. 1 ] room;
    ref lazy SessionTrack [0..1] sessionTrack;
    ref lazy ScheduleItem [0 .. 1] scheduleItem;

}

@CachePolicy("-R-")
@TargetOS("Qt")
dto ScheduleItem {
    domainKey int sessionId;
    var bool isBreak;
```

```
    var bool isLunch;
    var bool isEvent;
    var bool isRegistration;

    ref lazy Session [1] session;
}

@TargetOS("Qt")
dto Favorite {
    domainKey int sessionId;
    ref lazy Session [1] session;
}

@TargetOS("Qt")
dto Bookmark {
    domainKey int sessionId;
    ref lazy Session [1] session;
}

@TargetOS("Qt")
@CachePolicy("-R-T-")
// session lists created on demand
dto SessionLists {
    domainKey QString uuid;
    // implicitly ref lazy
    var Session [ 0 .. * ] scheduledSessions;
    var Session [ 0 .. * ] sameTimeSessions;
    var Session [ 0 .. * ] specialTimeSessions;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto SessionLink {
    domainKey QString uuid;
    var QString url;
    var QString title;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto Speaker {
    domainKey int speakerId;
    var bool isDeprecated;
    // 'SCHUL'
    var QString sortKey;
```

```
    // 'S'
    var QString sortGroup;
    var QString name;
    var QString publicName;
    var QString bio;

    ref lazy SpeakerImage [0..1] speakerImage;
    // implicitly ref lazy
    // sort day, start
    var Session [0 .. *] sessions;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto SpeakerImage {
    domainKey int speakerId;
    var QString originImageUrl;
    var bool downloadSuccess;
    var bool downloadFailed;
    var bool inAssets;
    var bool inData;
    var QString suffix;
    // 0 .. 4
    var int maxScaleFactor;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto SessionTrack {
    domainKey int trackId;
    var QString name;
    var bool inAssets;
    // implicitly ref lazy
    // sort day start
    var Session [0 .. *] sessions;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto Day {
    domainKey int id;
    // 0..6
    var int weekDay;
    @DateFormatString("yyyy-MM-dd")
    var Date conferenceDay;
```

```dtos
    // implicitly ref lazy
    // sort day start
    var Session [0 .. *] sessions;
}


// A P I   data delivered as payload from server via REST API
@CachePolicy("-R-T-")
@TargetOS("Qt")
dto SessionAPI {
    @ForeignPropertyName("id")
    domainKey int sessionId;
    var QString title;
    var QString subtitle;
    var QString description;
    @ForeignPropertyName("type")
    var QString sessionType;
    @DateFormatString("HH:mm")
    @ForeignPropertyName("start")
    var Time startTime;
    var QString duration;
    @ForeignPropertyName("abstract")
    var QString abstractText;
    var QString room;
    var QString track;
    @ForeignPropertyName("persons")
    var PersonsAPI [0 .. *] presenter;
    @ForeignPropertyName("links")
    var SessionLinkAPI [0 .. *] sessionLinks;
}
@CachePolicy("-R-T-")
@TargetOS("Qt")
dto PersonsAPI {
    @ForeignPropertyName("id")
    domainKey int speakerId;
}
@CachePolicy("-R-T-")
@TargetOS("Qt")
dto SessionLinkAPI {
    domainKey QString uuid;
    var QString url;
    var QString title;
}
@CachePolicy("-R-T-")
@TargetOS("Qt")
dto SpeakerAPI {
```

```
        domainKey int id;
        @ForeignPropertyName("first_name")
        var QString firstName;
        @ForeignPropertyName("last_name")
        var QString lastName;
        @ForeignPropertyName("full_public_name")
        var QString publicName;
        @ForeignPropertyName("abstract")
        var QString bio;
        var QString avatar;
    }

}
```