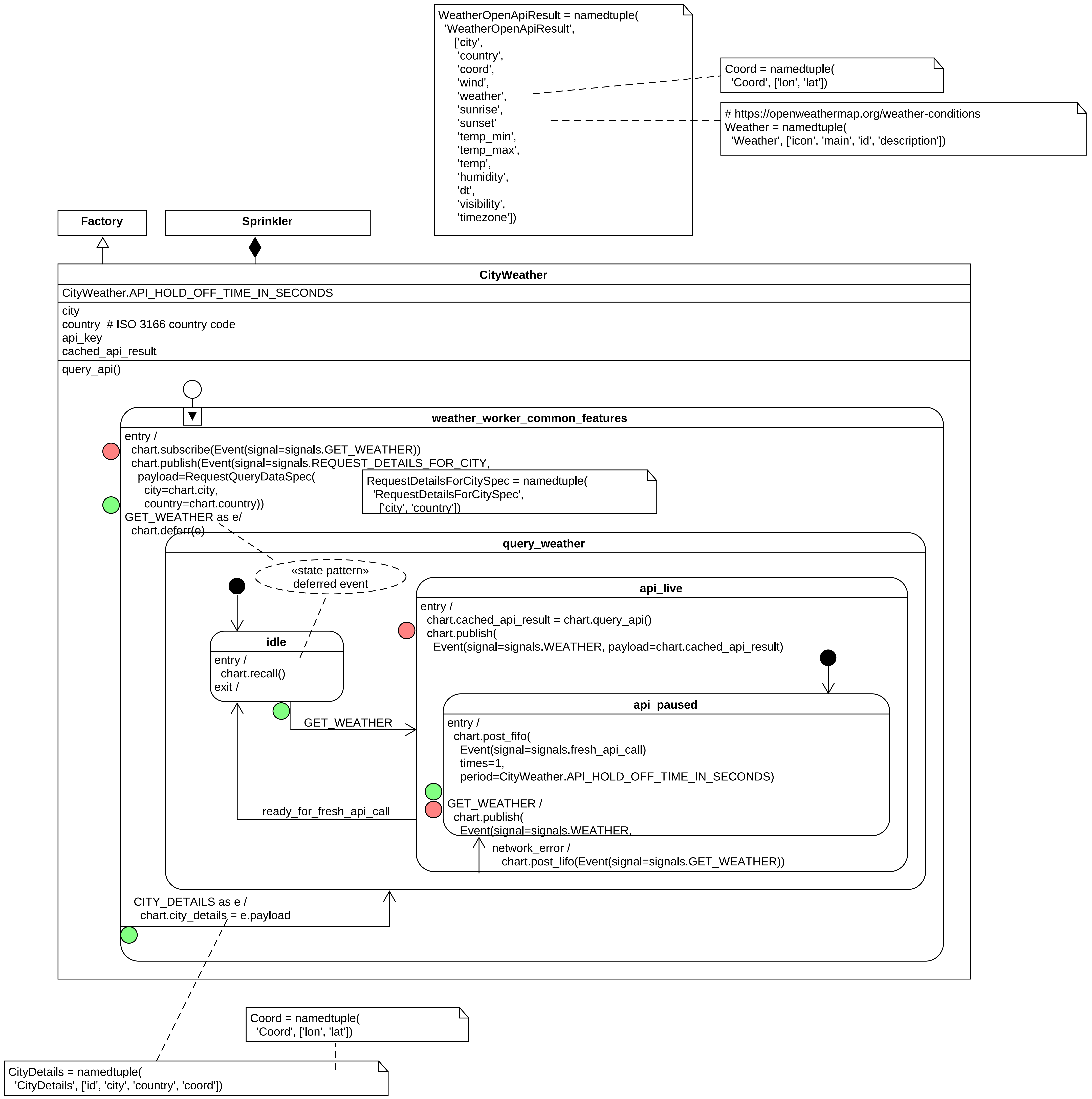

```
WeatherOpenApiResult = namedtuple(
  'WeatherOpenApiResult',
     ['city',
      'country',
      'coord',
      'wind',
      'weather',
      'sunrise',
      'sunset'
      'temp_min',
      'temp_max',
      'temp',
      'humidity',
      'dt',
      'visibility',
      'timezone'])
```

```
Coord = namedtuple(
  'Coord', ['lon', 'lat'])
```

```
# https://openweathermap.org/weather-conditions
Weather = namedtuple(
  'Weather', ['icon', 'main', 'id', 'description'])
```

```
CityWeather.API_HOLD_OFF_TIME_IN_SECONDS
```

```
city
country  # ISO 3166 country code
api_key
cached_api_result
```

```
query_api()
```

```
entry /
  chart.subscribe(Event(signal=signals.GET_WEATHER))
  chart.publish(Event(signal=signals.REQUEST_DETAILS_FOR_CITY,
    payload=RequestQueryDataSpec(
      city=chart.city,
      country=chart.country))
GET_WEATHER as e/
  chart.deferr(e)
```

```
RequestDetailsForCitySpec = namedtuple(
  'RequestDetailsForCitySpec',
    ['city', 'country'])
```

```
entry /
  chart.recall()
exit /
```

```
entry /
  chart.cached_api_result = chart.query_api()
  chart.publish(
    Event(signal=signals.WEATHER, payload=chart.cached_api_result)
```

```
entry /
  chart.post_fifo(
    Event(signal=signals.fresh_api_call)
    times=1,
    period=CityWeather.API_HOLD_OFF_TIME_IN_SECONDS)

GET_WEATHER /
  chart.publish(
    Event(signal=signals.WEATHER,
```

```
network_error /
   chart.post_lifo(Event(signal=signals.GET_WEATHER))
```

```
CITY_DETAILS as e /
  chart.city_details = e.payload
```

```
Coord = namedtuple(
  'Coord', ['lon', 'lat'])
```

```
CityDetails = namedtuple(
  'CityDetails', ['id', 'city', 'country', 'coord'])
```