

# Aplikace pro sběr hokejové telemetrie pro taktickou přípravu zápasu

Šimon Drda

21. prosince 2025

# Obsah

|   |           |
|---|-----------|
| <b>1 Analýza existujících řešení</b>                          | <b>5</b>  |
| 1.1 NHL Edge . . . . .  | 5         |
| 1.1.1 Techniky sběru dat NHL Edge . . . . .                   | 7         |
| 1.1.2 Komerční využití NHL Edge . . . . .                     | 8         |
| 1.1.3 Využití NHL Edge trenéry a analytiky . . . . .          | 9         |
| 1.1.4 Kritické zhodnocení veřejné prezentace NHL Edge . . . . | 9         |
| 1.2 Catapult Sports . . . . .                                 | 10        |
| 1.2.1 Aplikace pro zpracování dat Catapult Sports . . . . .   | 10        |
| 1.2.2 Techniky sběru dat Catapult Sports . . . . .            | 10        |
| 1.3 Kinexon Sports . . . . .                                  | 11        |
| 1.4 Drive Hockey . . . . .                                    | 12        |
| 1.5 Helios . . . . .  | 12        |
| 1.6 Srovnání řešení . . . . .                                 | 13        |
| 1.7 Návrh nových funkcionalit . . . . .                       | 14        |
| <b>2 Byznys a funkční požadavky</b>                           | <b>14</b> |
| 2.1 Byznys požadavky . . . . .                                | 14        |
| 2.1.1 Hlavní cíle . . . . .                                   | 14        |
| 2.1.2 Stakeholdeři . . . . .                                  | 15        |
| 2.2 Funkční požadavky . . . . .                               | 15        |
| 2.2.1 Sběr a přenos dat . . . . .                             | 15        |
| 2.2.2 Správa senzorů a hráčů . . . . .                        | 15        |
| 2.2.3 Správa událostí . . . . .                               | 15        |
| 2.2.4 Vizualizace v reálném čase . . . . .                    | 15        |
| 2.2.5 Animované přehrávání zápasu . . . . .                   | 16        |
| 2.2.6 Statistiky . . . . .                                    | 16        |
| 2.2.7 Heatmapy a analytické výpočty . . . . .                 | 16        |
| 2.2.8 Klíčové momenty zápasu . . . . .                        | 16        |
| 2.2.9 Export videa . . . . .                                  | 17        |
| 2.2.10 Taktická příprava a anotace . . . . .                  | 17        |
| 2.3 Nefunkční požadavky . . . . .                             | 17        |
| 2.3.1 Výkon . . . . .   | 17        |
| 2.3.2 Spolehlivost . . . . .                                  | 17        |
| 2.3.3 Uživatelská přívětivost . . . . .                       | 17        |
| 2.3.4 Bezpečnost . . . . .                                    | 17        |

|   |           |
|---|-----------|
| <b>3 Teorie lokalizačních senzorů</b>                         | <b>18</b> |
| 3.1 Two-way ranging . . . . .                                 | 18        |
| 3.2 Výpočet polohy tagu . . . . .                             | 19        |
| 3.2.1 Příklad multilaterace se čtyřmi kotvami . . . . .       | 20        |
| <b>4 Implementace lokalizačních senzorů</b>                   | <b>20</b> |
| 4.1 Přehled použitých komponent . . . . .                     | 20        |
| 4.2 Vývojové prostředí . . . . .                              | 21        |
| 4.3 Algoritmus výpočtu polohy tagu . . . . .                  | 22        |
| 4.4 Centrální kotva . . . . .                                 | 22        |
| 4.5 Rizika použití UWB technologie pro senzory . . . . .      | 23        |
| 4.5.1 Interference . . . . .                                  | 23        |
| 4.5.2 Zakrytí senzoru tělem hráče . . . . .                   | 23        |
| 4.5.3 Problémy s odrazy . . . . .                             | 24        |
| <b>5 Implementace klientské aplikace</b>                      | <b>25</b> |
| 5.1 Výběr platformy pro klientskou aplikaci . . . . .         | 25        |
| 5.2 Struktura a rozsah aplikace . . . . .                     | 26        |
| 5.2.1 Komunikační vrstva . . . . .                            | 27        |
| 5.2.2 Vrstva datového modelu . . . . .                        | 27        |
| 5.2.3 Analytická vrstva . . . . .                             | 27        |
| 5.2.4 Vizualizační vrstva . . . . .                           | 28        |
| 5.2.5 Aplikační logika . . . . .                              | 29        |
| 5.2.6 Návrh tříd a modelů . . . . .                           | 29        |
| 5.3 Vývojové prostředí . . . . .                              | 32        |
| 5.4 Závěr implementace klientské aplikace . . . . .           | 33        |
| <b>6 Testování</b>  | <b>33</b> |
| 6.1 Cíle testování . . . . .                                  | 33        |
| 6.2 Testování systému pro snímání dat . . . . .               | 33        |
| 6.2.1 Rozsah testování . . . . .                              | 33        |
| 6.2.2 Testování lokalizační přesnosti . . . . .               | 33        |
| 6.2.3 Testování rychlosti přenosu dat . . . . .               | 35        |
| 6.3 Testování klientské aplikace . . . . .                    | 35        |
| 6.3.1 Funkční testování . . . . .                             | 35        |
| 6.3.2 Testování datového modelu a analytické vrstvy . . . . . | 36        |
| 6.3.3 Integrační testování . . . . .                          | 36        |
| 6.3.4 Testování výkonu a odezvy aplikace . . . . .            | 36        |

|          |  |           |
|----------|--|-----------|
| 6.3.5    | Testování uživatelských scénářů . . . . .          | 37        |
| <b>7</b> | <b>Návrh harmonogramu implementace</b>             | <b>37</b> |
| 7.1      | Harmonogram vývoje lokalizačního systému . . . . . | 38        |
| 7.2      | Harmonogram vývoje klientské aplikace . . . . .    | 39        |
| 7.3      | Shrnutí harmonogramu . . . . .                     | 39        |

# 1 Analýza existujících řešení

V této kapitole provedeme rešerši existujících projektů a produktů, které se zabývají problematikou sběru hokejové telemetrie a jejího využití v praxi. Cílem je identifikovat mezery na trhu, porozumět potřebám uživatelů a zjistit, jaké funkce a vlastnosti jsou nejvíce žádané. Budeme se věnovat nejen samotným systémům, ale také jejich reálnému uplatnění.

## 1.1 NHL Edge

NHL Edge je pokročilý systém sběru dat vyvinutý americkou firmou SMT, která pronikla s toutu problematikou i do ostatních významných soutěží amerického fotbalu, basketbalu či motosportu.

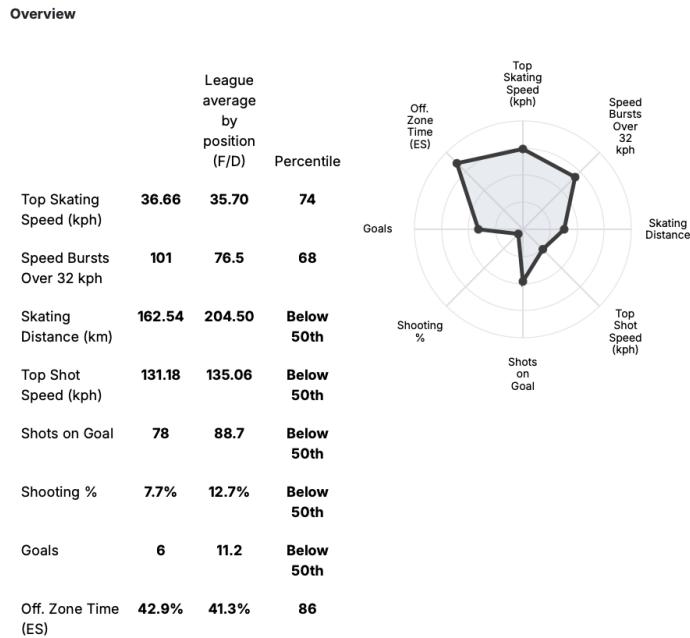
Tento systém je založený na čipech umístěných ve dresech hráčů a v puku, které sledují jejich pohyb a interakce na ledě. Data jsou následně zpracována a prezentována veřejnosti, trenérům a analytikům prostřednictvím specializovaných softwarů.

Od roku 2023 je Edge dostupný i pro veřejnost prostřednictvím aplikace NHL nebo webové stránky, což umožňuje fanouškům přístup k detailním statistikám a analýzám konkrétních údajů. Jelikož chceme, aby náš projekt též poskytoval část pro veřejnost, analýza toho, co Edge nabízí a jaké funkce by mohly být přínosné i pro náš projekt, je nezbytná. Přístupné jsou pouze informace o hráčích v poli, brankářích a týmech, nikoliv o konkrétních zápasech nebo situacích. Informace jsou kombinací dat z čipů a manuálního označování událostí během zápasu.

- **Statistiky hráčů v poli:** Prezentační software umožňuje vybrat tým, následně hráče a zobrazit tyto statistiky: Největší vyvinutá rychlosť, počet zrychlení nad 32 km/h, celková ujetá vzdálenost, nejrychlejší střela, počet střel, úspěšnost střelby, góly a procentuální pohyb v útočném pásmu.

Dále jsou k dispozici detailnější statistiky bruslení, které uvádějí statistiky zrychlení v různých rychlostních pásmech.

Navíc lze zobrazit i podrobný rozbor ujeté vzdálenosti. Celkový počet ujetých kilometrů, počet ujetých kilometrů za 60 minut (standardní doba zápasu), nejlepší zápas z pohledu ujeté vzdálenosti a nejlepší třetina.



Obrázek 1: Ukázka zobrazení statistik hráče v systému NHL Edge. Zdroj: Mathew Barzal, NHL Edge

Uživatelé zde najdou také o rychlosti střel, které jsou opět rozděleny do rychlostních pásem. Nejrychlejší střela, průměrná rychlosť střelby a počet střel v jednotlivých pásmech.

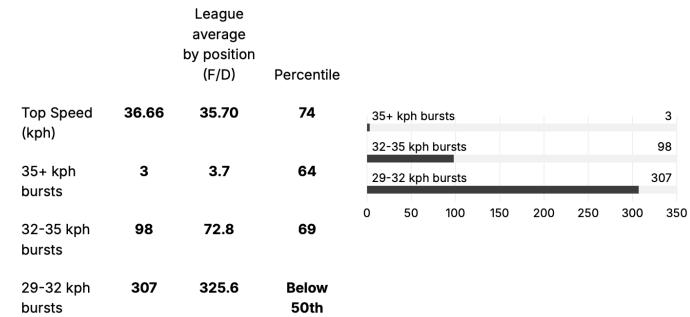
Další zajímavou informací je rozřazení střel podle místa jejich provedení. Na heatmapě je přehledně vidět, odkud hráč nejčastěji střílí. (Obrázek 3)

Poslední statistikou je rozdělení tzv. zone time neboli času stráveného v jednotlivých pásmech. Uživatel zde vidí, kolik procent času hráč strávil v obranném, neutrálním a útočném pásmu.

- **Statistiky brankářů:** U golmánu nemáme žádná data, která by pocházela z čipů, ale pouze klasické statistiky jako počet zákroků, procento úspěšnosti, počet inkasovaných gólů a počet vyhraných zápasů. Tyto statistiky se rozdělují do podrobnějších kategorií, ale z důvodu nevyužití čipů je nepovažuji za relevantní pro náš projekt.

#### Skating Speed

Max speed measures the maximum sustained skating speed a player has achieved during the current season. Bursts measure the number of times a skater achieved a sustained speed above a given threshold. Results divided by positions groups (forwards, defensemen).



Obrázek 2: Ukázka zobrazení statistik bruslení v systému NHL Edge. Zdroj: Mathew Barzal, NHL Edge

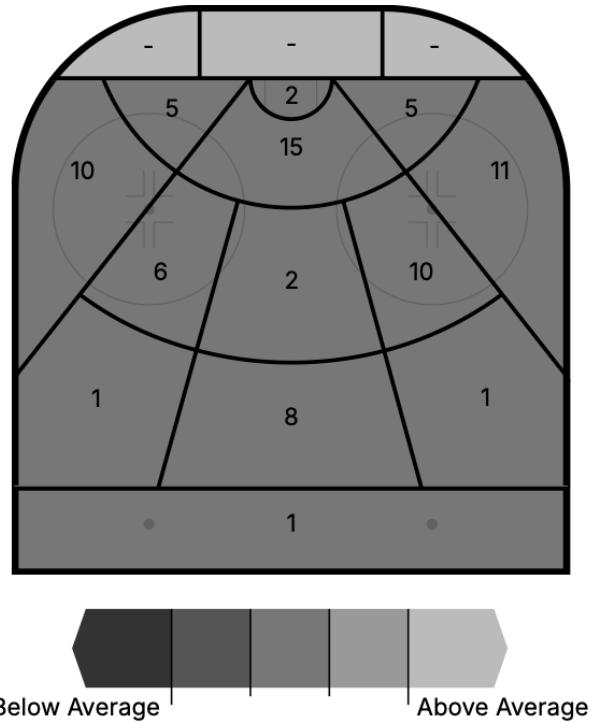
- **Statistiky týmů:** Statistiky týmů zahrnují obdobné statistiky, které jsme viděli v hráčské kategorii, ale jsou prezentovány z pohledu celého týmu.

#### 1.1.1 Techniky sběru dat NHL Edge

Jelikož tato práce má za účel navrhnout i konkrétní techniku sběru potřebných údajů, nejen jejich vyhodnocování, je vhodné stručně nastínit metody sběru.

Systém Edge využívá čipy umístěné v dresech hráčů a v puku, jejichž signál je zachycován sítí 14 až 18 infračervených kamer rozmístěných kolem ledové plochy. Pohyb puku je zaznamenáván 60krát za sekundu, zatímco pohyb hráčů 15krát za stejný časový úsek.

Jedním z problémů této metody je potenciální blokace signálu těly hráčů, což může způsobovat výpadky dat. SMT však tento nedostatek minimalizuje umístěním kamer do větší výšky a jejich dostatečným počtem. Je relevantní uvést problém detekovaný u ženského hokeje – culíky hráček částečně stínily čipy v dresech, což vedlo k chybovosti. Tento nedostatek byl vyřešen úpravou umístění čipů.



Obrázek 3: Ukázka zobrazení lokalit střel v systému NHL Edge. Zdroj: Mathew Barzal, NHL Edge

### 1.1.2 Komerční využití NHL Edge

Kromě statistik dostupných v aplikaci NHL Edge pro veřejnost, jedna z oblastí, kde se aktivně využívají sesbíraná data, je televizní přenos. Zejména se jedná o zobrazení rychlosti střel, polohy hráčů, délky jejich střídání a jejich jména.

Zajímavostí je NHL Hockeyverse, což je animovaný 3D model zápasu, který využívá sesbíraná data. NHL takto prezentuje několik vybraných zápasů pro dětské diváky.

### 1.1.3 Využití NHL Edge trenéry a analytiky

Data z NHL Edge nejsou trenérům a analytikům poskytována přímo, ale prostřednictvím aplikací a nástrojů partnerů NHL, především SMT a SAP. Oficiálním rozhraním pro týmy je *SAP Coaching Insights App for iPad*, která kombinuje data z NHL Edge (sledování hráčů a puku) s oficiálními herními statistikami NHL. Podle informací SMT lze data vizualizovat v reálném čase na taktických tabulích (*whiteboard*), přehrávat záznamy z předchozích zápasů a analyzovat je pomocí nástrojů, jako jsou heatmapy střel, trajektorie pohybu hráčů a puku nebo individuální i týmové statistiky. Tyto funkce mají pro trenéry zásadní význam v taktické přípravě – například na základě heatmap soupeře lze identifikovat nebezpečné střelecké zóny, nebo podle analýzy trajektorií plánovat forčeking a obranné formace.

### 1.1.4 Kritické zhodnocení veřejné prezentace NHL Edge

Ačkoli je NHL Edge technologicky vyspělý systém s obrovským množstvím dat, jeho využitelnost má několik zásadních omezení:

- **Zaměření na fanoušky:** Veřejná verze je primárně orientována na prezentaci dat pro fanoušky a média, nikoli na potřeby trenérů. Taktické funkce jsou dostupné pouze v interních nástrojích, ke kterým běžní uživatelé nemají přístup.
- **Chybějící kontext:** Systém pro veřejnost poskytuje velké množství číselných údajů, ale jen málo z nich je zasazených do kontextu konkrétní herní situace. Například víme, že hráč dosáhl maximální rychlosti 36 km/h, ale není zřejmé, jak tato informace souvisí s efektivitou jeho presinku či úspěšností vstupu do útočného pásma.
- **Neúplná data o brankářích:** Statistika brankářů je velmi omezená a nezahrnuje detailní prostorové analýzy zákroků, což je pro přípravu přesilovek či střeleckých strategií zásadní. Tato mezera výrazně snižuje celkovou hodnotu systému pro taktické plánování.
- **Absence zápasů:** Veřejná verze neposkytuje možnost analyzovat konkrétní zápasy nebo situace, což omezuje její využitelnost pro hlubší analýzu a taktický rozbor veřejnosti.

Z těchto důvodů lze říci, že NHL Edge představuje vynikající zdroj surových dat, ale pro veřejnost postrádá většího využití než jen pro zábavu fanoušků.

## 1.2 Catapult Sports

Dalším významným hráčem na poli sportovní telemetrie je společnost Catapult Sports, která nabízí řešení pro různé sporty včetně hokeje. Přímé využití pro taktické plánování je ale omezené, zaměřují se spíše na kondici a prevenci zranění. Jejich systém využívá GPS a akcelerometry, umístěné do speciální podprsenky. Oproti NHL Edge Catapult neumožňuje přístup k datům veřejnosti.

### 1.2.1 Aplikace pro zpracování dat Catapult Sports

Catapult poskytuje pro hokej 3 druhy softwaru. Tím, který zpracovává nasnímaná data z podprsenek, je Vector. Dále provozují softwarové systémy Thunder a Pro Video, které slouží pro video analýzu a vizualizaci dat.

Pro Video nabízí možnost připojení telemetrických dat ze systému Vector, je tedy možné synchronizovat videozáznam s aktuálními metrikami. Dle dostupných zdrojů zde ale není detailní zaměření na zobrazování dat pro hokej, nevyskytuje se zde žádne 2D hřiště, které by pro trenéra poskytovalo stejný pohled jako taktická tabule. To znamená, že ačkoliv Catapult dobře podporuje kondiční přípravu a prevenci zranění, jeho přímý přínos pro taktickou přípravu zápasu je omezený.

### 1.2.2 Techniky sběru dat Catapult Sports

Samotné senzory fungují, stejně jako u NHL Edge, na principu LPS, tedy lokalizačního systému vnitřního prostoru nazývaný ClearSky, který využívá rádiové vlny. Systém je schopný snímat polohu hráče s přesností na 10 cm a zaznamenává data s frekvencí 10 Hz. Senzory jsou vybaveny akcelerometry, gyroskopy a magnetometry, které umožňují měřit různé aspekty pohybu hráče, jako je rychlosť, zrychlení, skoky a změny směru. Z dostupných informací plyne, že zařízení Vector 8 (nejnovější

model modulu, který se umístí do podprsenky) disponuje bezdrátovou komunikací UWB i Bluetooth. Pro venkovní snímání oproti minulé verzi modulu Vector S7 je možné využít i GNSS (Global Navigation Satellite System - obecný termín pro globální navaigacní satelitní systémy, které pomocí satelitů určují polohu zařízení na Zemi).

Pokud se zaměříme na realizaci sběrového systému, který data z UWB sensoru přijímá, zjistíme, že ClearSky vyžaduje tzv. kotvy ClearSky [10], které se musí nainstalovat v místě sledování. Existuje možnost využít pouze jednu kotvu pro sledování inerciálních dat (zrychlení, skoky, změny směru), ale nikoliv pro určení polohy. Zajímavostí je, že kotvy pracující v pásmu UWB (ultra-wideband) na pásmech 4GHz a 6,5GHz, jsou rušeny 5G signálem a WiFi 6E. To při rostoucím trendu nasazování těchto technologií bude v blízké budoucnosti představovat problém pro firmu Catapult Sports, která bude muset své produkty přizpůsobit. Pražská O2 arena, kde hraje své domácí zápasy tým HC Sparta Praha, zavedla podporu 5G sítě v roce 2024 [8], což pro případnou implementaci systému Catapult Sports omezuje výběr pásem pouze na 6,5GHz.

### 1.3 Kinexon Sports

Třetím konkurentem kterého uvedeme do povědomí je německá firma Kinexon Sports sídlící v Mnichově. Ze zmíněných tří firem je nejméně známa, ale i přesto její služby využívají věhlasné kluby jako Eisbären Berlin nebo New York Rangers. Kinexon nabízí podobný systém jako Catapult, tedy senzory připojené přes UWB s 10 centimetrovou přesností a latencí 100ms. Narozdíl od jejich druhého produktu Kinexon Perform IMO, které slouží k plug and play sledování omezených metrik bez nutnosti přizpůsobení prostředí, vyžaduje Kinexon Perform LPS, který je nutný pro hokej, podobné kotvy jako v případě Catapult Sports. Teoreticky by podle nich měly stačit 3 kotvy, ale doporučují se minimálně 4.

Kinexon nabízí software, který je kombinací vizualizace dat vykreslených na whiteboard a manuálně zadávaných událostí. Z pohledu taktické přípravy může Kinexon trenérům poskytnout detailní pohybová data hráčů, ale bez sofistikovaného softwaru pro vizualizaci herních situací zůstává jeho využitelnost pro přípravu strategie omezená.

## 1.4 Drive Hockey

Drive Hockey představuje moderní systém určený pro sledování hráčů pomocí nositelných senzorů a zpracování telemetrických dat v reálném čase. Jeho cílem je poskytnout trenérům i hráčům nástroje pro analýzu výkonu, herního stylu a efektivity pohybu. Podle výrobce je možné systém zprovoznit během 15 minut, přičemž dokáže sledovat až 30 hráčů na ledě a dalších 50 na střídačce. Drive Hockey Player Tracking System využívá senzory umístěné na výstroji hráčů, které přenášejí data o rychlosti, akceleraci, trajektorii pohybu a interakci hráčů s pukem.

Z pohledu funkcí se Drive Hockey přibližuje našemu návrhu více než většina konkurence. Nabízí přehledy individuálních i týmových výkonů, heatmapy, přehrávání klíčových situací (*replay*) a možnost označování důležitých momentů v zápase. Platforma rozlišuje přístupová práva – hráči mohou přistupovat ke svým vlastním datům, trenéři mají k dispozici přehled celého týmu. Tento princip odpovídá i návrhu naší aplikace, která rovněž počítá s uživatelským rozhraním pro oba typy uživatelů.

Z technického hlediska není zcela jasné, zda Drive Hockey využívá UWB, nebo jinou formu bezdrátové lokalizace. Výrobce však uvádí, že systém umožňuje sledování v reálném čase a zobrazení dat v grafické podobě – například prostřednictvím 2D pohledu na led. Nabízí rovněž export dat a statistik, i když bez plně integrovaného video exportu.

## 1.5 Helios

Helios Hockey je systém zaměřený spíše na individuální výkon hráče než na týmovou taktiku. Senzor s názvem *Core Performance Sensor* se připevňuje na ramenní chránič a měří bruslení, výbušnost, zrychlení a další parametry. Helios Hockey poskytuje aplikaci pro mobilní zařízení, která hráči umožňuje přehledně sledovat vývoj svých metrik v čase a porovnávat je s ostatními. Systém také synchronizuje senzorová data s videozáznamy, čímž lze analyzovat konkrétní momenty zápasu.

Zaměření Heliosu je výhradně individuální — neumožňuje zobrazení celé ledové plochy, pohybu všech hráčů ani taktickou vizualizaci. Pro tréninkové účely, zejména pro zlepšování bruslařských a střeleckých dovedností, však poskytuje užitečné přehledy o rychlosti, stabilitě a rov-

nováze. Tím se výrazně liší od řešení, která pracují s kompletní týmovou dynamikou (např. NHL Edge nebo Kinexon).

## 1.6 Srovnání řešení

Na základě provedené rešerše lze jednotlivé systémy srovnat podle technických i funkčních aspektů. Tabulka 1 uvádí přehled klíčových vlastností pěti vybraných produktů ve srovnání s návrhem našeho systému.

Tabulka 1: Srovnání existujících systémů pro sběr a analýzu hokejové telemetrie

| Funkce / systém                | NHL Edge                   | Catapult             | Kinexon                     | Helios                    | Drive Hockey            |
|--------------------------------|----------------------------|----------------------|-----------------------------|---------------------------|-------------------------|
| UWB lokalizace hráčů/puků      | Ano (čip + IR kamery)      | Ano (ClearSky LPS)   | Ano (UWB kotvy)             | Ne                        | Nejasné (ano – senzory) |
| IMU / akcelerometry            | Ano                        | Ano                  | Ano                         | Ano                       | Ano                     |
| Zaměření                       | Fanouškovská prezentace    | Kondiční příprava    | Kondiční a výkonová analýza | Individuální výkon        | Taktická analýza        |
| Registrace hráčů / senzorů     | Ne (interní)               | Ano                  | Ano                         | Ano                       | Ano                     |
| 2D vizualizace polohy          | Cástečně (NHL Hockeyverse) | Ne                   | Cástečně                    | Ne                        | Ano                     |
| Heatmapy                       | Ano (střely)               | Ne                   | Cástečně                    | Ne                        | Ano                     |
| Klíčové momenty / replay       | Ne                         | Ne                   | Cástečně                    | Ano (video + data)        | Ano                     |
| Přístup hráče k vlastním datům | Ne                         | Ano                  | Ano                         | Ano                       | Ano                     |
| Export videa / animace         | Ne                         | Cástečně (Pro Video) | Ne                          | Ano (video synchronizace) | Ano (replay/export)     |
| Taktická příprava zápasu       | Omezeně (interní aplikace) | Ne                   | Omezeně                     | Ne                        | Ano                     |

Z tabulky vyplývá, že:

- **NHL Edge** představuje technologicky nejvyspělejší systém v oblasti kombinace čipů a kamer, ale jeho veřejná verze je zaměřená spíše na fanoušky.
- **Catapult** a **Kinexon** dominují v oblasti kondiční přípravy, nikoli taktické analýzy.
- **Helios** se specializuje na individuální měření výkonu hráče, nikoli týmovou dynamiku.
- **Drive Hockey** je ze všech komerčně dostupných produktů nejblíže navrhovanému řešení – kombinuje metriky výkonu, replay, heatmapy a uživatelský přístup hráčů i trenérů.

Z tabulky lze vyčíst, že je nutné při návrhu našeho systému přínést výhody navíc, které by zlepšily uživatelský komfort a funkčnost. V následujících 2 kapitolách budou postupně navrhovány technologické a uživatelské řešení s ohledem na zjištěné nedostatky existujících systémů, zejména Drive Hockey.

## 1.7 Návrh nových funkcionalit

Oproti systému *Drive Hockey*, který se zaměřuje především na analýzu výkonu a statistické přehledy, je cílem navrhovaného řešení nabídnout trenérům i hráčům interaktivní nástroj pro taktickou přípravu [4, 5]. Následující tabulka shrnuje nové funkctionality, které rozšiřují možnosti systému.

Tabulka 2: Srovnání funkcionalit s Drive Hockey

| Funkce                            | Popis   | Rozdíl oproti Drive Hockey   |
|-----------------------------------|---|--|
| Taktický režim                    | Interaktivní 2D vizualizace pohybu hráčů a puku s možností simulace a po-zastavení průběhu [11].          | Drive Hockey nabízí pouze přehrávání bez interakce a anotací.      |
| Whiteboard s Apple Pencil         | Možnost zakreslování taktických situaci přímo do vizualizace, tvorba tréninkových scénářů [1].            | Drive Hockey nepodporuje nativní kreslení ani scénářové plánování. |
| Automatická detekce událostí (AI) | Heuristická analýza pro automatické rozpoznávání typů akcí a taktických vzorců [4].                       | Drive Hockey vyžaduje ruční označení.                              |
| Offline režim s lokální databází  | Možnost plného využití aplikace bez připojení během zápasu, následná synchronizace po návratu online [3]. | Drive Hockey vyžaduje trvalé připojení.                            |

## 2 Byznys a funkční požadavky

### 2.1 Byznys požadavky

Z rešerše vyplýva, že pro funkční originalitu našeho projektu musíme oproti původnímu zadání přidat několik byznys a funkčních požadavků.

#### 2.1.1 Hlavní cíle

- poskytovat přesná polohová data hráčů a puku,
- podporovat taktické rozhodování a přípravu na zápas,
- generovat statistiky, heatmapy a klíčové momenty,
- umožnit vizuální analýzu pomocí 2D animací,
- sjednotit práci se senzory, hráči a událostmi,
- fungovat na iPadOS s real-time odezvou.

### 2.1.2 Stakeholdeři

- **Hráč** – registrace senzoru, přístup k vlastním výkonům.
- **Trenér** – vizualizace, taktické nástroje, export videí.
- **Analytik** – práce s heatmapami, statistikami a detekcemi.
- **Technický personál** – konfigurace kotev a síťové infrastruktury.
- **Organizace** – dlouhodobý přehled o výkonech hráčů.

## 2.2 Funkční požadavky

### 2.2.1 Sběr a přenos dat

- příjem dat z UWB senzorů přes WebSocket,
- formát zprávy obsahuje `tagId`, `x`, `y`, `timestamp`,
- real-time zpracování s minimální latencí.

### 2.2.2 Správa senzorů a hráčů

- hráč se může registrovat pomocí svého senzoru,
- trenér přiřazuje senzory a hráče k událostem,
- hráč má přístup pouze ke svým vlastním zápasům.

### 2.2.3 Správa událostí

- vytváření událostí typu *zápas* nebo *tréning*,
- přidání hráčů a jejich senzorů,
- ukládání trajektorií a detekovaných situací.

### 2.2.4 Vizualizace v reálném čase

- vizualizace aktuálních pozic hráčů a puku,
- pozastavení přenosu, zpětné přehrání,
- identifikace hráčů a jejich pohybu na hřišti.

### **2.2.5 Animované přehrávání zápasu**

- přehrání celého zápasu ve formě 2D animace,
- časová osa, rychlosť přehrávání,
- opakované přehrání vybraných úseků.

### **2.2.6 Statistiky**

#### **Individuální statistiky hráče**

- počet střel a jejich úspěšnost,
- rychlostní maxima,
- čas na ledě a počet střídání,
- celková trajektorie a překonaná vzdálenost.

#### **Týmové statistiky**

- držení puku,
- kolize,
- intenzita tlaku v jednotlivých zónách.

### **2.2.7 Heatmapy a analytické výpočty**

- generování heatmap střel, pohybu a držení puku,
- filtrování podle hráče a časového úseku,
- detekce klíčových momentů pomocí heuristik nebo CoreML.

### **2.2.8 Klíčové momenty zápasu**

- zobrazení gólu, střel, kolizí, změn momentum a úniků,
- kliknutím na moment se přehraje související sekvence.

### **2.2.9 Export videa**

- výběr časového úseku,
- generování 2D animace do video formátu,
- možnost sdílení na iPadOS.

### **2.2.10 Taktická příprava a anotace**

- kreslení do hřiště pomocí Apple Pencil,
- ukládání taktických scénářů,
- anotace animací a videí.

## **2.3 Nefunkční požadavky**

### **2.3.1 Výkon**

- zobrazení polohy do 50 ms od přijetí dat,
- 60 FPS při animaci zápasu.

### **2.3.2 Spolehlivost**

- schopnost pracovat offline,
- odolnost vůči výpadkům spojení.

### **2.3.3 Uživatelská přívětivost**

- optimalizace pro iPadOS,
- jednoduché ovládání během zápasu.

### **2.3.4 Bezpečnost**

- přístup k datům podmíněný registrací senzoru,
- oddělení dat hráčů a trenérů.

## 3 Teorie lokalizačních senzorů

Po zvážení faktoru nedostupnosti dat z UWB senzorů z profesionálních souťží i jakýchkoliv dat, které by se daly pro semestrální práci použít, musíme přistoupit k implementaci vlastního řešení, které nám poskytne data poloh hráčů a puku, na kterém poté budeme moci stavět datové operace a vizualizace.

Celý systém po vzoru již vzniklých komerčních řešení založíme na pevných bodech tzv. kotvách a na hybných bodech, senzorech umístěných v objektech měnících polohu - hráči a puk. Pro návrh fyzických zařízení nebo-li senzorů využijeme lokalizační moduly **DWM1000** ve spolupráci s populární vývojovou deskou **ESP32**, které jsou pro naše potřeby maximálně postačující. Využijeme jejich schopnost mezi sebou provozovat tzv. **two-way ranging** [6] pro měření vzdálenosti.

### 3.1 Two-way ranging

Two-way ranging umožňuje určit vzdálenost mezi dvěma UWB moduly pomocí měření časů odeslání a přijetí několika zpráv. Celý proces se skládá ze tří kroků: senzor (tag) odešle zprávu **POLL**, kotva odpoví zprávou **RESPONSE** a nakonec tag odešle zprávu **FINAL**. Obě strany si ukládají přesná časová razítka odeslání a přijetí těchto zpráv a z nich je poté možné spočítat čas letu signálu. [2]

Princip měření vzdálenosti pomocí two-way ranging vychází z přesného časování odeslaných a přijatých UWB rámců. Během výměny zpráv **POLL**, **RESPONSE** a **FINAL** jsou na obou stranách zaznamenána časová razítka odeslání a příjmu. Jednotlivé naměřené časové intervaly lze vyjádřit jako

$$d = c \cdot \frac{(T_{\text{resp}} - T_{\text{poll}}) - (T_{\text{final}} - T_{\text{tx}})}{2},$$

kde

- $T_{\text{poll}}$  je čas odeslání zprávy **POLL** tagem,
- $T_{\text{resp}}$  je čas přijetí zprávy **RESPONSE** tagem,
- $T_{\text{tx}}$  je čas odeslání zprávy **RESPONSE** kotvou,
- $T_{\text{final}}$  je čas přijetí zprávy **FINAL** kotvou,

- $c$  je rychlosť šírenia elektromagnetické vlny (rychlosť svetla).

Časové intervaly ( $T_{\text{resp}} - T_{\text{poll}}$ ) a ( $T_{\text{final}} - T_{\text{tx}}$ ) obsahují mimo samotného času letu signálu také neznámé zpoždění způsobené zpracováním rámců na protější straně. Odečtením těchto dvou intervalů se toto zpoždění vzájemně eliminuje a výsledkem je dvojnásobek skutečného času letu signálu. Po vydelení dvěma tedy získáme čas letu signálu v jednom směru, který po vynásobení rychlosť svetla  $c$  udává výslednou vzdáenosť mezi tagem a kotvou.

### 3.2 Výpočet polohy tagu

Pokud získáme informaci o vzdáenosťi tagu vůči  $n$  kotvám, kdy  $n \geq 3$ , můžeme využít principy multilaterace. Multilaterace je technika zjištění polohy subjektu v prostoru. Pro zjištění dvoudimenzionálních souřadnic v teorii stačí 3 kotvy, avšak v praxi se může vyskytnout řada problémů jako např. stínění. Tyto problémy vyplývají z výzkumné části projektu.

Pokračujeme výpočtem polohy pomocí multilaterace [9]. Každá kotva s danou souřadnicí  $(x_i, y_i)$  a naměřenou vzdáenosťí  $d_i$  definuje kružnici, na jejímž obvodu se může tag nacházet. Pro kotvy  $i = 1 \dots n$  platí obecná nelineární rovnice

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2.$$

Přímé řešení těchto rovnic není triviální, a proto se rovnice převádějí na lineární tvar. Zvolíme první kotvu jako referenční a odečteme její rovnici od rovnic ostatních kotev. Tím získáme linearizovaný vztah

$$2(x_1 - x_i)x + 2(y_1 - y_i)y = d_1^2 - d_i^2 + x_i^2 - x_1^2 + y_i^2 - y_1^2, \quad i = 2 \dots n.$$

Soustavu lze zapsat maticově jako

$$A = \begin{bmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) \\ 2(x_1 - x_3) & 2(y_1 - y_3) \\ \vdots & \vdots \\ 2(x_1 - x_n) & 2(y_1 - y_n) \end{bmatrix}, \quad b = \begin{bmatrix} d_1^2 - d_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ d_1^2 - d_3^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2 \\ \vdots \\ d_1^2 - d_n^2 + x_n^2 - x_1^2 + y_n^2 - y_1^2 \end{bmatrix}.$$

Výslednou polohu tagu získáme metodou nejmenších čtverců:

$$\begin{bmatrix} x \\ y \end{bmatrix} = (A^\top A)^{-1} A^\top b.$$

### 3.2.1 Příklad multilaterace se čtyřmi kotvami

Uvažujme čtyři kotvy se známými souřadnicemi

$$A_1 = (0, 0), A_2 = (10, 0), A_3 = (0, 10), A_4 = (10, 10).$$

Nechť skutečná poloha tagu je

$$p = (4, 3).$$

Vzdálenosti od tagu ke kotvám jsou

$$d_1 = 5, \quad d_2 \approx 6,7082, \quad d_3 \approx 8,0623, \quad d_4 \approx 9,2195.$$

Po dosazení do linearizovaných rovnic získáme:

$$\begin{aligned} 20x = 80 &\Rightarrow x = 4, \\ 20y = 60 &\Rightarrow y = 3. \end{aligned}$$

Čtvrtá kotva zde slouží k ověření správnosti výpočtu – při dosazení vypočtených souřadnic do rovnice pro  $A_4$  je rovnice splněna. V případě měření zatíženého šumem se použije maticové řešení  $(A^\top A)^{-1} A^\top b$ , které poskytne nejlepší odhad polohy.

Tento postup lze aplikovat pro libovolný počet kotev  $n \geq 3$ , přičemž více kotev zpravidla vede k vyšší přesnosti výsledku.

## 4 Implementace lokalizačních senzorů

### 4.1 Přehled použitých komponent

V této podkapitole si kráce shrneme komponenty, které budou využity pro implementaci kotev a hráčských senzorů. Soubor komponent podléha neutálemu výzkumu. Momentálně jsem v očekávání většiny zmíněných součástek.

Celé zařízení bude postaveno na již zmiňovaném DWM1000 od firmy Decawave. Jedná se o lokalizační modul pracující ve standardu IEEE 802.15.4a na frekvencích od 3,5 GHz do 6,5 GHz [7]. Tento modul bude spolupracovat s populární vývojovou deskou ESP32, která bude mít na starost řízení samotného lokalizačního modulu a komunikaci s ostatními kotvami přes Wi-Fi.

Jelikož naše ESP32 nemá napájecící obvod, pro připojení k Li-Ion baterii také musíme přidat USB-C nabíječku baterií s ochranou a také tedy samotnou baterii.

Jediným rozdílem mezi návrhem kotev a tagů by měla být integrace malého informačního OLED displeje, který by sloužil pro příjemnější nastavování samotného kotevního systému.

## 4.2 Vývojové prostředí

Vývoj a testování hardwarových komponent kotvy a hráčského tagu probíhá s využitím následujícího softwarového a vývojového ekosystému:

- **PlatformIO** – hlavní vývojové prostředí pro tvorbu firmwaru, integrované v editoru VS Code.
- **Visual Studio Code** – víceúčelový editor sloužící jako IDE pro firmware ESP32 i práci s knihovnami.
- **Arduino Core for ESP32** – nízkoúrovňová vrstva a knihovny pro zjednodušený vývoj nad čipy ESP32.
- **FreeRTOS** – real-time operační systém integrovaný v ESP-IDF, využívaný pro práci s vlákny a plánováním úloh.
- **ESP-IDF** – oficiální vývojový framework společnosti Espressif pro pokročilejší správu periferií, komunikace a výkonovou optimalizaci.
- **DWM1000 Arduino Library** – knihovna pro řízení UWB modulu Decawave DWM1000 a implementaci komunikace dle IEEE 802.15.4a.
- **U8g2** – grafická knihovna pro obsluhu OLED displeje použitého v kotvách.
- **ESP Async WebServer / WiFi Library** – nástroje pro síťovou komunikaci, nastavování kotev a management přes Wi-Fi.
- **KiCad** – návrh zapojení, schémat a DPS jednotlivých hardwarových modulů.
- **Serial Monitor a Logic Analyzer** – ladicí nástroje pro analýzu datové komunikace mezi ESP32 a DWM1000.

### 4.3 Algoritmus výpočtu polohy tagu

Na uvedených vzorcích založíme základní algoritmus pro výpočet polohy tagu. Algoritmus vychází z předpokladu, že v jednom měřicím cyklu získáme vzdálenosti  $d_i$  mezi tagem a jednotlivými kotvami, přičemž každá kotva má známé pevné souřadnice  $(x_i, y_i)$ . Následně využijeme linearizovanou formu rovnic multilaterace a získáme tak odhad dvoudimenzionální pozice tagu.

Základní postup lze shrnout do následujících kroků:

1. Tag zahájí měřicí cyklus vysláním rámce **POLL**.
2. Jednotlivé kotvy odpovídají postupně v přidělených časových slotech, čímž se zajistí, že nedojde ke kolizi UWB rámci.
3. Každá kotva provede two-way ranging a spočítá vzdálenost  $d_i$  mezi sebou a tagem.
4. Hodnoty vzdáleností a souřadnice kotev jsou odeslány master kotvě.
5. Master kotva sestaví matici  $A$  a vektor  $b$  dle vztahů popsaných v předchozí kapitole.
6. Pomocí metody nejmenších čtverců master kotva vyhodnotí odhad polohy tagu:

$$\begin{bmatrix} x \\ y \end{bmatrix} = (A^\top A)^{-1} A^\top b.$$

7. Výsledná poloha je následně přenesena do nadřazené aplikace, jejíž implementace je vysvětlena v následující kapitole implementační části.

Takto navržený algoritmus umožňuje stanovit polohu tagu i za přítomnosti měřicího šumu, protože metoda nejmenších čtverců minimalizuje chybu způsobenou nepřesnými vzdálenostmi. Počet kotev může být libovolně vyšší než tři, což dále zvyšuje robustnost výpočtu.

### 4.4 Centrální kotva

Pro komunikaci s externími aplikacemi, například vizualizační aplikací na tabletech iPadOS, jejíž implementace bude detailně rozepsána v další kapitole, master kotva vystavuje jednoduchý server běžící na námi použitém mikrokontroléru ESP32. Vhodným řešením je využití protokolu *WebSocket*,

který umožňuje obousměrnou komunikaci v reálném čase bez nutnosti opanovaného dotazování. Master kotva po každém výpočtu polohy odešle zprávu v JSON formátu ve tvaru

```
{  
    "tagId": "player_1",  
    "x": 4.03,  
    "y": 2.98,  
    "timestamp": 1717001244  
}
```

Klientská aplikace na iPadu se k master kotvě připojuje jako WebSocket klient a v reálném čase přijímá aktualizace polohy tagu. Tato architektura umožňuje okamžitou vizualizaci pohybu hráče nebo puku na ledové ploše bez znatelného zpoždění.

## 4.5 Rizika použití UWB technologie pro senzory

### 4.5.1 Interference

Jedním z potencionálních rizikem použití UWB technologie je rušení s jinými bezdrátovými systémy. V rešeršní části jsme uvedli problém, kdy UWB na pásmech 4GHz a 6,5GHz je rušeno 5G signálem a WiFi 6E. S narůstajícím výskytem těchto technologií i do prostředí zimních stadionů, je nutno s tímto rizikem počítat. K řešení by byl nutný výzkum vhodných pásem uvnitř hokejových hal.

### 4.5.2 Zakrytí senzoru tělem hráče

Dalším rizikem je ztráta signálu v případě nevhodného zakrytí senzoru. Tento problém lze dle mého eliminovat umístěním senzoru po vzoru konkurenčních komerčních řešení v horní části hráčské výstroje, popřípadě na helmě.



Obrázek 4: Umístění senzoru v dresu hráče severoamerické NHL

#### 4.5.3 Problémy s odrazy

V hokejové hale nalezneme spoustu kovových konstrukcí (části mantinelů, branky), která nám mohou ztěžovat hladký průběh multilaterace. Zdrojem našeho tvrzení je experimentální studie zaměřená na funkčnost UWB ve vnitřních prostorách [12], která prezentuje výrazné zmenšení přesnosti při výskytu kovových konstrukcí a lidí samotných. Ze studie též plyne doporučení umístění kotev na nejvyšší možné místo. To jen potvrzuje chování komerčních řešení. Ani my bychom tedy neměli mít v plánu umístit kotvy na hokejové mantinely, ale spíše ke střeše samotné haly.

Pro testování našeho systému by tedy byla ideální hala poskytující možnost pohybu ve střešních prostorách. V Česku existují několik arén, které vhodný přístup poskytují. Pro vizualizaci si uvedeme zimní stadion v severočeském Mostě, kde je možný pohyb v těsné blízkosti střechy a byl by ideálním kandidátem pro pilotní provoz našeho systému.



Obrázek 5: Zimní stadion Most - bude nahrazen vlastně pořízeným obrázkem

## 5 Implementace klientské aplikace

Klientská aplikace představuje uživatelské rozhraní celého lokalizačního systému a poskytuje trenérům, hráčům i analytikům nástroje pro práci s daty získanými z UWB senzorů umístěných v dresech a puku. Je navržena pro platformu iPadOS, která nabízí vysoký výkon, kvalitní dotykové prostředí a možnosti práce s Apple Pencil. Hlavním cílem je zajištění real-time vizualizace, analytických výpočtů, taktických nástrojů a správy zápasů a tréninků.

### 5.1 Výběr platformy pro klientskou aplikaci

Výběr cílové platformy byl založen na osobním pozorování, že se v těsném prostředí hokejové střídačky, i po většinou těsných prostorech hokejových hal, těžce využívají větší zařízení jako např. notebook. Tablety jsou díky své tenkosti tedy vhodným kandidátem pro toto prostředí. Oproti mobilním telefonům zasa naopak poskytují větší výkon srovnatelný s notebooky, lepší výdrž a větší displej.

Zvolení **iPadOS** oproti Androidu bylo poté založeno na mé osobní preferenci a vlastniství zařízení, na kterém je možné vyvíjet. Zároveň je zde mnohobásobně menší počet možných výrobců - pouze značka Apple - a řad cílových zařízení, což velice usnadňuje vývoj.

## 5.2 Struktura a rozsah aplikace

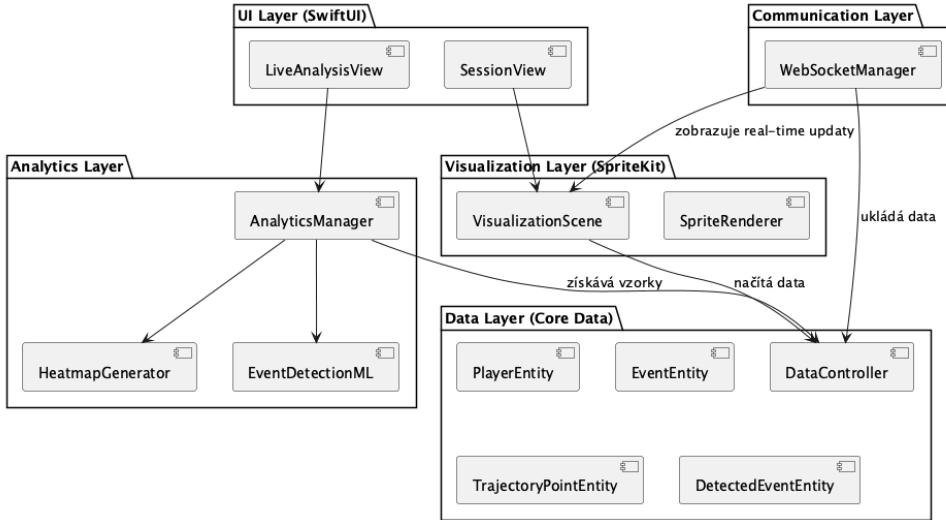
Aplikace je navržena jako modulární systém složený z několika vrstev a funkčních celků. Každý modul má jasně definovanou odpovědnost, čímž je zajištěna snadná rozšiřitelnost, testovatelnost a udržitelnost kódu.

Celkovou strukturu aplikace tvoří zejména:

- **Komunikační vrstva** – odpovídá za příjem dat z centrální kotvy přes WebSocket.
- **Vrstva datového modelu** – správa entit, ukládání trajektorií, správa uživatelů, zápasů a událostí pomocí Core Data.
- **Analytická vrstva** – výpočet statistik, generování heatmap a detekce klíčových momentů pomocí heuristik a CoreML.
- **Vizualizační vrstva** – vykreslení hokejového hřiště, pohybu hráčů, puku a všech interaktivních prvků pomocí SpriteKit.
- **Taktická vrstva** – nástroje pro vizuální analýzu a tvorbu scénářů, anotace pomocí Apple Pencil.
- **Aplikační logika** – správa událostí (zápasů, tréninků), práce s uživateli, přehrávání zápasů a export videa.

Tento rozsah by měl pokrývat vše, co jsme si vydefinovali v našich byznysových a funkčních požadavcích.

### Diagram architektury klientské aplikace



Obrázek 6: UML diagram architektury klientské aplikace.

### 5.2.1 Komunikační vrstva

Tato vrstva se postará o komunikaci mezi centrální kotvou a naší aplikací pomocí technologie WebSocket. Využijeme jeho minimální prodlevu mezi změnou a doručením dat. Na straně klientské aplikace využijeme nativní knihovnu Network, která poskytuje rozhrání pro práci s toutu technologií. Součástí vrstvy bude i vydefinování **Data Transfer Objects** pro přenos jednotlivých entit ze sběrného zařízení. Formát zpráv odpovídá struktuře `tagId, x, y, timestamp`, definované ve funkčních požadavcích.

### 5.2.2 Vrstva datového modelu

Vrstva datového modelu bude definovat jednotlivé CoreData entity, jejich Repository nadstavby a správu jejich životního cyklu.

### 5.2.3 Analytická vrstva

Analytická vrstva představuje klíčovou část navrhovaného systému, neboť přímo zajišťuje naplnění hlavních funkčních požadavků definovaných v zadání práce. Jejím úkolem je transformovat surová data získaná z UWB senzorů umístěných v dresech hráčů a v puku na interpretovatelné analytické

výstupy využitelné pro statistické vyhodnocení, vizualizaci a taktickou analýzu zápasů a tréninků.

Vrstva zpracovává časové a prostorové trajektorie jednotlivých senzorů a nad těmito daty provádí výpočty statistik individuálních i týmových výkonů. Na základě těchto výpočtů umožňuje systém zobrazovat statistické výběry, jako jsou nejtvrzší střela, nejrychlejší bruslař, čas strávený na ledě, počet střel, úspěšnost přihrávek nebo počet střídání jednotlivých hráčů, čímž přímo naplňuje požadavek na získání přehledů individuálních výkonů hráčů.

Dále analytická vrstva slouží k tvorbě prostorových analýz ve formě heatmap, které jsou generovány z vybraných statistických údajů, například z míst zakončení střel nebo pohybu hráčů v jednotlivých herních situacích. Tyto výstupy umožňují vizuální analýzu rozložení hry na hřišti a podporují rozhodovací proces trenérů a analytiků.

Významnou součástí analytické vrstvy je rovněž detekce klíčových momentů zápasu, jako jsou góly, kolize hráčů, únikové situace nebo změny tempa a herního momenta. Detekce těchto událostí je realizována kombinací heuristických pravidel a do budoucna případně metod strojového učení využívajících framework CoreML (např. že bych pokračoval s tímto tématem u diplomové práce). Identifikované klíčové momenty slouží jako vstup pro přímý přechod k vybraným situacím v rámci přehrávání zápasu.

Analytická vrstva zároveň poskytuje podklad pro přehrávání celého zápasu ve formě 2D animace a pro generování vybraných úseků této animace do podoby videa určeného pro další práci videotrenérů. V tomto kontextu analytická vrstva zajišťuje přípravu a agregaci dat, zatímco samotná vizualizace a export jsou realizovány v dalších vrstvách systému.

#### 5.2.4 Vizualizační vrstva

Vizualizační vrstva slouží k přehlednému a interaktivnímu zobrazení dat zpracovaných analytickou vrstvou. Je navržena pro platformu **iPadOS** a využívá práci v režimu celé obrazovky, která nám umožní ideálně využít plochu poskytovanou velikostí tabletů.

Hlavním vizuálním prvkem je 2D zobrazení hokejového hřiště, nad kterým jsou pomocí frameworku SpriteKit vykreslovány trajektorie pohybu hráčů a puku a realizováno plynulé přehrávání zápasu nebo tréninku ve formě animace. Uživatel může prostřednictvím dotykového ovládání pracovat s časovou osou, přecházet ke klíčovým momentům zápasu a přiblížovat vybrané části hrací plochy.

Vizualizační vrstva umožňuje současné zobrazení více typů informací, jako jsou heatmapy, zvýraznění herních událostí nebo individuální trajektorie vybraných hráčů, přímo nad hrací plochou. Tímto způsobem podporuje detailní analýzu hry bez nutnosti přepínání mezi jednotlivými obrazovkami.

Vizualizační vrstva podporuje také interaktivní kreslení a anotace nad hrací plochou, včetně podpory Apple Pencil, pro účely taktické přípravy.

Technicky je vizualizační vrstva realizována kombinací frameworků SpriteKit a SwiftUI. Pracuje výhradně s výstupy analytické vrstvy.

### 5.2.5 Aplikační logika

Aplikační logika se stará o generální správu událostí, které seskupuje do projektů, jenž je možné vytvářet, ukládat a exportovat pro přenos do jiného zařízení. Je též zamýšlena jako centrální bod koordinace veškeré komunikace mezi jednotlivými vrstvami.

Aplikační logika zároveň zajišťuje přiřazení senzorů hráčům, správu uživatelských rolí (trenér/hráč) a omezení přístupu hráčů pouze k jejich vlastním událostem.

### 5.2.6 Návrh tříd a modelů

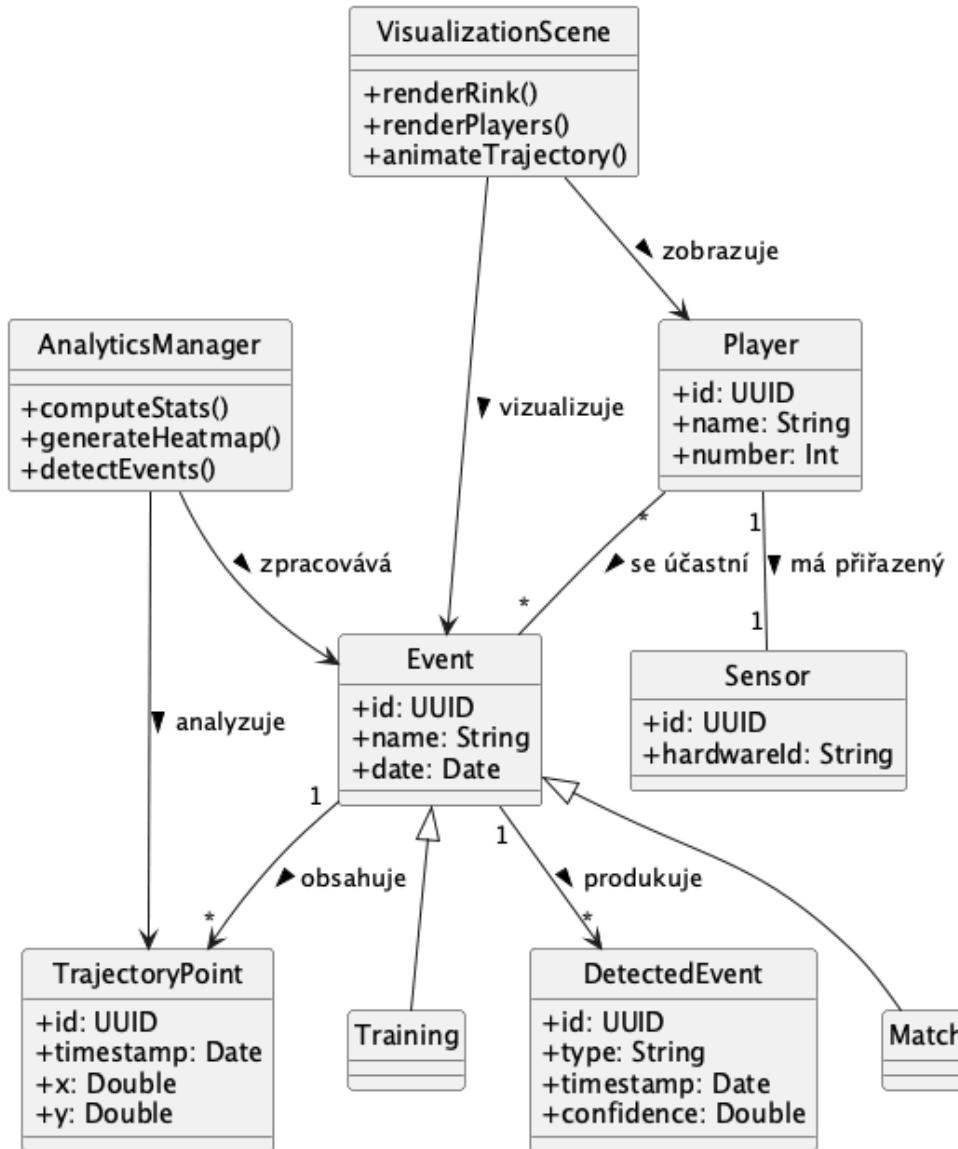
Následující část popisuje hlavní třídy a jejich vazby v podobě class diagramu. Diagram shrnuje strukturu datového modelu aplikace a vztahy mezi jednotlivými entitami.

#### Hlavní entity:

- **Player** – reprezentuje hráče, obsahuje identifikační údaje a vazbu na jeho senzor.
- **Sensor** – UWB tag, který generuje polohová data; je přiřazen hráči.
- **Event** – abstraktní entita reprezentující zápas nebo trénink.
- **Match** – podtyp *Event*, reprezentuje hokejový zápas.
- **Training** – podtyp *Event*, reprezentuje tréninkovou jednotku.
- **TrajectoryPoint** – jednotlivé polohové vzorky z UWB měření (x, y, timestamp).

- **DetectedEvent** – systémem automaticky nebo ručně označené klíčové momenty (góл, kolize, střela, ztráta puku).
- **AnalyticsManager** – modul pro statistiky, heatmapy a rozpoznávání událostí.
- **VisualizationScene** – SpriteKit scéna zodpovědná za vykreslení hřiště a animací.

### Diagram tříd



Obrázek 7: UML diagram tříd datového modelu klientské aplikace.

**Význam diagramu:** Diagram zachycuje základní datové entity aplikace a jejich vztahy:

- Každý hráč má jeden UWB senzor.

- Každá událost (zápas/trénink) obsahuje stovky až tisíce polohových vzorků.
- Detekované události jsou vždy svázány s konkrétním zápasem.
- Hráč může být účastníkem mnoha událostí.

Tato struktura odpovídá požadavkům na ukládání velkého množství dat a umožňuje efektivně generovat statistiky a vizualizace.

### 5.3 Vývojové prostředí

Aplikace je vyvíjena s využitím moderního softwarového stacku Apple:

- **Git a GitHub** – pro správu a verzování samotného kódu.
- **Xcode 12** – hlavní IDE pro vývoj iPadOS aplikace.
- **Swift 6.2** – programovací jazyk pro implementaci aplikační logiky.
- **SwiftUI** – deklarativní framework pro tvorbu uživatelského rozhraní.
- **SpriteKit** – 2D engine pro vykreslení hokejového hřiště a animací hráčů.
- **Core Data** – lokální databáze pro ukládání událostí, hráčů a trajektorií.
- **CoreML a CreateML** – strojové učení pro automatickou detekci taktických momentů.
- **Apple Pencil API** – implementace taktického whiteboardu a anotací.
- **WebSocket** – komunikační kanál mezi iPadem a centrální kotvou.

Vývoj probíhá na macOS, který poskytuje plnou podporu pro komplikaci a testování iPadOS aplikací. Aplikace je testována v simulátoru iPadOS i na fyzickém zařízení, zejména kvůli latenci WebSocket přenosu a interakcím s Apple Pencil.

## 5.4 Závěr implementace klientské aplikace

Tato část práce představila implementaci klientské aplikace z hlediska struktury, architektury, návrhu tříd a použitého vývojového prostředí. Popsané moduly tvoří pevný základ pro interaktivní taktickou přípravu, vizualizaci pohybu hráčů a pokročilou analýzu založenou na UWB datech. Díky modulární architektuře a využití moderních nástrojů ekosystému Apple je aplikace připravena na další rozšiřování a nasazení v tréninkové i zápasové praxi.

# 6 Testování

## 6.1 Cíle testování

Cílem testování je ověřit správnost návrhu a implementace systému sběru hokejové telemetrie a její následné vyhodnocení. Zaměříme se především na kvalitu přenosu a přesnost našeho UWB systému a ověření klientské aplikace tak, aby odpovídala požadavkům reálného provozu v praxi.

## 6.2 Testování systému pro snímání dat

V první řadě se zaměříme na systém založený na UWB pro sběr dat. V jeho teoretické části jsme si uvedli několik možných problémů, se kterým se můžeme při vývoji potýkat.

### 6.2.1 Rozsah testování

V rámci finančních možností musíme testování provést v rozložení 4 kotev určených pro multilateraci a jednoho hráčského lokalizačního tagu. I přesto, že se počet kotev zdaleka nerovná počtu v profesionálních projektech, poskytne nám dostatek dat pro vyhodnocení přibližné polohy hráče v různých částech hřiště. Posledním bodem systému bude centrální kotva určená pro sběr dat a následnou komunikaci s klientskou aplikací.

### 6.2.2 Testování lokalizační přesnosti

Lokalizační systém je testován pomocí jednoho hráče, který se pohybuje v rámci předem definovaného scénáře. Tímto scénářem myslíme testování tří hlavních druhu pohybu, které budou podrobeny vizuální analýze správnosti

naměřených dat včetně natáčení videozáznamu pro možnost zpětné analýzy. Za ztěžejní body ideálního systému lokalizace považuji přesnost údajů vzdáleností z kotvy a správnost algoritmu, pomocí kterého určíme polohu hráče.

**Statická poloha** Hráč se postaví na předem určené referenční body na hřišti - body pro vhazování. Těch nalezneme na celém hřišti 9.

**Pohyb po přímce** Hráč se bude pohybovat mezi stanovenými body po přímce po celém hřišti.

- Levý horní bod pro vhazování v obranném pásmu – Pravý horní bod pro vhazování v obranném pásmu
- Pravý horní bod pro vhazování v obranném pásmu – Pravý horní bod pro vhazování ve středním pásmu
- Pravý horní bod pro vhazování ve středním pásmu – Levý horní bod pro vhazování ve středním pásmu
- Levý horní bod pro vhazování ve středním pásmu – Středový bod pro vhazování
- Středový bod pro vhazování – Levý dolní bod pro vhazování ve středním pásmu
- Levý dolní bod pro vhazování ve středním pásmu – Pravý dolní bod pro vhazování ve středním pásmu
- Pravý dolní bod pro vhazování ve středním pásmu – Pravý dolní bod pro vhazování v obranném pásmu
- Pravý dolní bod pro vhazování v obranném pásmu – Levý dolní bod pro vhazování v obranném pásmu

**Nahodilý pohyb po celé ploše hřiště** V posledním scánaři nejvíce využijeme vizuální porovnání, kdy hráč dostane za úkol libovolně se pohybovat po hřišti. Zejména dostane za úkol dělat náhlé změny směru, zrychlení, prudké brždění a přechod z jízdy pozadu do jízdy vpřed a *vice versa*.

### **6.2.3 Testování rychlosti přenosu dat**

Rychlosť přenosu dat je testována měřením časového rozdílu mezi:

- okamžikem odeslání měření z lokalizačního tagu,
- okamžikem jeho zobrazení v klientské aplikaci.

Testování je opět prováděno s jedním hráčem, aby bylo možné přesně vyhodnotit latenci bez vlivu zátěže způsobené více současně přenášenými datovými toky a abychom vyhověli rozpočtu našeho projektu. Vyhodnocována je:

- průměrná latence přenosu,
- stabilita přenosu v čase,
- výskyt ztracených nebo opožděných paketů.

## **6.3 Testování klientské aplikace**

Testování klientské aplikace je zaměřeno na ověření správnosti zpracování příchozích telemetrických dat, jejich vizualizace a celkové použitelnosti aplikace pro účely taktické přípravy zápasu. Testy jsou navrženy tak, aby reflektovaly reálné pracovní scénáře trenérů a analytiků.

### **6.3.1 Funkční testování**

Funkční testování ověřuje, že jednotlivé funkce klientské aplikace pracují v souladu s funkčními požadavky definovanými v kapitole 2. Testovány jsou zejména následující oblasti:

- navázání spojení se zdrojem telemetrických dat,
- správné načítání a ukládání datových záznamů,
- zobrazení aktuální polohy hráče na 2D modelu hřiště,
- vykreslování trajektorie pohybu hráče v čase,
- generování heatmapy pohybu,
- přehrávání zaznamenaného úseku pohybu.

Každá funkce je ověřena pomocí předem definovaných vstupních dat při testu systému pro sběr dat a očekávaných výstupů.

### **6.3.2 Testování datového modelu a analytické vrstvy**

Datový model a analytická vrstva jsou testovány odděleně od vizualizační části aplikace. Testování zahrnuje:

- správné mapování příchozích dat a jejich filtrování na interní datové struktury,
- korektní časovou synchronizaci dat,
- výpočty rychlosti, zrychlení a ujeté vzdálenosti,
- správne filtrování příchozích dat,
- přípravu dat pro vizualizační prvky.

Testy jsou prováděny pomocí syntetických dat s přesně definovanými hodnotami.

### **6.3.3 Integrační testování**

Integrační testování ověřuje spolupráci jednotlivých vrstev klientské aplikace a jejich napojení na zdroj telemetrických dat. Testované scénáře zahrnují:

- plynulé zpracování kontinuálního datového toku,
- správné předávání dat mezi komunikační, analytickou a vizualizační vrstvou,
- reakci aplikace na krátkodobý výpadek dat,
- opětovné navázání spojení bez nutnosti restartu aplikace.

Zvláštní pozornost je věnována chování aplikace při zpoždění dat, které může nastat při bezdrátovém přenosu přes WebSocket technologii nastat.

### **6.3.4 Testování výkonu a odezvy aplikace**

Testování výkonu je zaměřeno na ověření, že aplikace je schopna zpracovávat příchozí data v reálném čase bez znatelného zpoždění uživatelského rozhraní. Vyhodnocováno je:

- plynulost vykreslování pohybu hráče,

- odezva aplikace na uživatelské akce (posun v čase, zakreslení informací do hríště),
- stabilita aplikace při delším běhu.

Výsledky testování budou zobrazeny kvanitativně v tabulkové formě v porovnání s referenčními hodnotami.

### 6.3.5 Testování uživatelských scénářů

Uživatelské testování simuluje reálné použití aplikace trenérem nebo analytikem při přípravě na zápas. Testované scénáře zahrnují:

- sledování pohybu hráče v reálném čase,
- přehrávání konkrétní herní situace,
- analýzu pohybu pomocí heatmapy,
- identifikaci klíčových momentů pohybu.

Testování budeme provádět manuálně s cílem ověřit srozumitelnost rozhraní a přehlednost prezentovaných informací.

## 7 Návrh harmonogramu implementace

Tato kapitola se zaměřuje na návrh časového harmonogramu implementace navrhovaného systému pro sběr a analýzu hokejové telemetrie. Harmonogram popisuje jednotlivé fáze vývoje, jejich časovou náročnost a vzájemné závislosti. Cílem je navrhnout realistický a přehledný plán realizace jak hardwarové části systému, tak klientské aplikace určené pro taktickou přípravu hokejového utkání.

Harmonogram je rozdělen do dvou hlavních částí:

- vývoj lokalizačního systému založeného na UWB technologii,
- vývoj klientské aplikace pro platformu iPadOS.

Časové odhady jsou uvedeny v týdnech a vycházejí z rozsahu implementace, technické náročnosti jednotlivých úloh a z předpokladu vývoje jednou osobou, tedy mnou.

## 7.1 Harmonogram vývoje lokalizačního systému

Vývoj lokalizačního systému tvoří technologický základ celého řešení. Zahrnuje návrh hardwaru kotev a senzorů, implementaci komunikačních mechanismů, výpočet polohy hráčů a puku a zajištění přenosu dat do klientské aplikace v reálném čase. Harmonogram reflektuje postup od analytické fáze přes návrh a implementaci až po testování funkčnosti systému.

Tabulka 3: Harmonogram vývoje lokalizačního systému

| Fáze         | Činnost                         | Popis  | Odhad trvání |
|--------------|---------------------------------|--|--------------|
| Analýza      | Upřesnění požadavků             | Specifikace požadavků na přesnost lokalizace, latenci přenosu dat, počet kotev a způsob jejich vzájemné komunikace               | 1 týden      |
| Návrh        | Návrh hardwaru kotev            | Návrh zapojení lokalizačního modulu DWM1000, mikrokontroléru ESP32, napájecího obvodu a případného OLED displeje                 | 1 týden      |
| Návrh        | Topologie systému               | Návrh rozmístění kotev v prostoru haly, určení centrální (master) kotvy a definice datových toků mezi jednotlivými prvky systému | 0,5 týdne    |
| Implementace | Firmware kotev                  | Implementace two-way ranging, časování měření a synchronizace kotev  | 2 týdny      |
| Implementace | Komunikace mezi kotvami         | Přenos naměřených vzdáleností z jednotlivých kotev na centrální kotvu prostřednictvím bezdrátové sítě                            | 1 týden      |
| Implementace | Výpočet polohy                  | Implementace algoritmu multilaterace a metody nejmenších čtverců na centrální kotvě  | 1 týden      |
| Implementace | WebSocket rozhraní              | Implementace WebSocket serveru na centrální kotvě pro přenos polohových dat do klientské aplikace                                | 0,5 týdne    |
| Integrace    | Napojení na klientskou aplikaci | Ověření správnosti přenosu JSON zpráv a jejich zpracování v klientské aplikaci   | 0,5 týdne    |
| Testování    | Přesnost a latence              | Testování lokalizační přesnosti systému a odezvy přenosu dat v reálném čase  | 1 týden      |

Celková doba vývoje lokalizačního systému je odhadována přibližně na

osm týdnů.

## 7.2 Harmonogram vývoje klientské aplikace

Vývoj klientské aplikace probíhá částečně paralelně s vývojem lokalizačního systému. V počátečních fázích jsou pro implementaci a testování využívána jednoduchá simulovaná telemetrická data, zatímco v pozdějších fázích dochází k přímému napojení na centrální kotvu a zpracování reálných dat v reálném čase.

Tabulka 4: Harmonogram vývoje klientské aplikace

| Fáze         | Činnost                        | Popis  | Odhad trvání |
|--------------|--------------------------------|--|--------------|
| Analýza      | Uživatelské scénáře            | Upřesnění způsobu práce trenéra, hráče a analytika s aplikací                  | 0,5 týdne    |
| Návrh        | Architektura aplikace          | Návrh vrstev aplikace, datového modelu a toků dat                              | 1 týden      |
| Implementace | Komunikační vrstva             | Implementace WebSocket klienta a zpracování příchozích telemetrických dat      | 1 týden      |
| Implementace | Datový model                   | Návrh a implementace Core Data entit pro ukládání událostí a trajektorií       | 1 týden      |
| Implementace | Analytická vrstva              | Implementace výpočtů statistik, generování heatmap a detekce klíčových momentů | 2 týdny      |
| Implementace | Vizualizační vrstva            | 2D vizualizace hokejového hřiště, animace pohybu hráčů a práce s časovou osou  | 2 týdny      |
| Implementace | Taktické nástroje              | Implementace anotací, kreslení pomocí Apple Pencil a tvorby taktických scénářů | 1 týden      |
| Integrace    | Napojení na lokalizační systém | Zpracování reálných dat z centrální kotvy v klientské aplikaci                 | 0,5 týdne    |
| Testování    | Funkční a výkonové testy       | Ověření odezvy aplikace, plynulosti animací a stability systému                | 1 týden      |

Celková doba vývoje klientské aplikace je odhadována přibližně na jedenáct týdnů.

## 7.3 Shrnutí harmonogramu

Navržený harmonogram implementace nám umožňuje postupně řidit vývoj celého systému v podobě metody vodopádu. I přes to, že vývoj může pro-

bíhat najednou, celkových 20 týdnů nám říká, že časové okno od března do odevzdání navazující bakalářské práce nemusí stačit.

## Odkazy

- [1] *Apple Human Interface Guidelines – iPadOS*. Oficiální návrhové principy pro optimalizaci uživatelského rozhraní pro iPadOS. Apple Design Team. 2025. URL: <https://developer.apple.com/design/human-interface-guidelines/designing-for-ipados> (cit. 27. 10. 2025).
- [2] *APS013: DW1000 and Two-Way Ranging*. Tech. zpr. Application Note APS013. Decawave Ltd., 2016. URL: <https://forum.qorvo.com/uploads/short-url/x34DrF7EW5fQP9wY3aNESqPKz8z.pdf>.
- [3] *Core Data Framework*. Framework pro správu lokální databáze a perzistence dat v iOS aplikacích. Apple Inc. 2025. URL: <https://developer.apple.com/documentation/coredata> (cit. 27. 10. 2025).
- [4] *Core ML Framework*. Framework pro integraci modelů strojového učení do nativních iOS aplikací s podporou výpočtů on-device. Apple Inc. 2025. URL: <https://developer.apple.com/documentation/coreml> (cit. 27. 10. 2025).
- [5] *Create ML*. Nástroj Apple pro trénování modelů strojového učení na macOS, umožňuje přípravu modelů pro použití v Core ML. Apple Inc. 2025. URL: <https://developer.apple.com/documentation/createml> (cit. 27. 10. 2025).
- [6] Decawave Ltd. *DWM1000 Module Datasheet (Version 1.6)*. Version 1.6. Decawave Ltd. Led. 2016. URL: [https://www.laskakit.cz/user/related\\_files/dwm1000-datasheet-v1-6.pdf](https://www.laskakit.cz/user/related_files/dwm1000-datasheet-v1-6.pdf).
- [7] LaskaKit. *Decawave DWM1000 – lokalizační modul*. 2025. URL: <https://www.laskakit.cz/decawave-dwm1000-lokalizacni-modul/> (cit. 26. 11. 2025).
- [8] O2 arena. *O2 spustilo 5G síť v O2 areně*. O2 arena. 2024. URL: <https://www.dsl.cz/clanky/o2-spustilo-5g-sit-v-o2-arene>.

- [9] Mathias Pelka. *Position Calculation with Least Squares based on Distance Measurements*. Tech. zpr. TR-2/2015. Accessed: 2025-11-25. Lübeck University of Applied Sciences, 2015. URL: [https://www.th-luebeck.de/fileadmin/media\\_cosa/Dateien/Veroeffentlichungen/Sammlung/TR-2-2015-least-squares-with-ToA.pdf](https://www.th-luebeck.de/fileadmin/media_cosa/Dateien/Veroeffentlichungen/Sammlung/TR-2-2015-least-squares-with-ToA.pdf).
- [10] *PŘEHLED KOTEVNÍHO HARDWARU CLEARSKY*. Catapult Sports. 2025. URL: <https://support.catapultsports.com/hc/en-us/articles/360000476776-ClearSky-Anchor-Hardware-Overview>.
- [11] *SpriteKit Framework*. Oficiální framework Apple pro 2D grafiku a animace, vhodný pro vizualizaci pohybových dat v reálném čase. Apple Inc. 2025. URL: <https://developer.apple.com/documentation/spritekit> (cit. 27. 10. 2025).
- [12] Jong-Hoon Youn a Yong K. Cho. „Performance Study of an Ultra-Wideband Indoor Localization and Asset Tracking System“. In: *Ultra Wideband*. Accessed: 2025-11-26. InTech, 2010. DOI: 10.5772/10055. URL: [https://www.researchgate.net/publication/221909161\\_Performance\\_Study\\_of\\_an\\_Ultra-Wideband\\_Indoor\\_Localization\\_and\\_Asset\\_Tracking\\_System](https://www.researchgate.net/publication/221909161_Performance_Study_of_an_Ultra-Wideband_Indoor_Localization_and_Asset_Tracking_System).