Energy Calculations for PTAC vs PTHP Systems

EDK Energy Insight Platform

September 2, 2025

1 Introduction

This document outlines the energy calculations for comparing Packaged Terminal Air Conditioner (PTAC) and Packaged Terminal Heat Pump (PTHP) systems, using the standardized variable names from our codebase.

2 PTAC System Calculations

2.1 Per-Unit Constants

The system uses fundamental constants for PTAC units in original units and their MMBtu equivalents:

- `annualUnitThermsHeatingPTAC`: 255 therms per year per unit for heating
- `annualUnitKwhCoolingPTAC`: 16,000 kWh per year per unit for cooling
- `annualUnitMMBtuHeatingPTAC`: 25.5 MMBtu per year per unit for heating
  - Converted from 255 therms × 0.1 MMBtu/therm = 25.5 MMBtu
- `annualUnitMMBtuCoolingPTAC`: 5.459427 MMBtu per year per unit for cooling
  - Converted from 16,000 kWh × 0.003412 MMBtu/kWh = 5.459427 MMBtu

2.2 Building-Level Calculations

For a building with multiple units, we calculate both MMBtu and original unit values:

2.2.1 MMBtu Building Totals

$$\texttt{annualBuildingMMBtuCoolingPTAC} = N_{ptacUnits} \times \texttt{annualUnitMMBtuCoolingPTAC} \tag{1}$$

$$\texttt{annualBuildingMMBtuHeatingPTAC} = N_{ptacUnits} \times \texttt{annualUnitMMBtuHeatingPTAC} \tag{2}$$

$$\texttt{annualBuildingMMBtuTotalPTAC} = \texttt{annualBuildingMMBtuCoolingPTAC} + \texttt{annualBuildingMMBtuHeatingPTAC} \tag{3}$$

3 PTHP System Calculations

3.1 Heating Energy Conversion with Building-Specific EFLH

For PTHP systems, the heating energy is calculated using building-specific Equivalent Full Load Hours (EFLH) derived from PLUTO data, combined with PTHP heating capacity and coefficient of performance:

3.1.1 Step 1: Calculate Annual Building kWh for PTHP Heating

The annual building heating consumption in kWh is calculated using the detailed equation:

$$\texttt{annualBuildingkWhHeatingPTHP} = \frac{\texttt{heatingCapacityPTHP}}{3.412} \times \frac{1}{\texttt{pthpCOP}} \times \text{EFLH} \times N_{ptacUnits} \tag{4}$$

Where:

- `heatingCapacityPTHP`: 8 KBtu (constant heating capacity per PTHP unit)
- 3.412: Conversion factor from KBtu to kW (kW per KBtu)
- `pthpCOP`: 1.51 (Coefficient of Performance for PTHP)
- EFLH: Equivalent Full Load Hours (building-specific, from PLUTO data)
- $N_{ptacUnits}$: Number of PTAC units to be replaced

3.1.2 Step 2: EFLH Lookup from PLUTO Data

The EFLH value is determined by building characteristics from PLUTO data using the following function:

```
export function getEFLHFromPluto(yearBuilt: number | string, floors: number | string): number {
  const [year, numFloors] = [Number(yearBuilt), Number(floors)];
  if (!Number.isFinite(year) || year <= 0 || !Number.isFinite(numFloors) || numFloors <= 0) {
    throw new Error(`Invalid inputs: yearBuilt=${yearBuilt}, floors=${floors}`);
  }

  const buildingType = numFloors <= 6 ? "LOW_RISE" : "HIGH_RISE";
  const constructionEra = year <= 1939 ? "PREWAR" :
                          year <= 1978 ? "PRE79" :
                          year <= 2006 ? "1979_2006" : "2007PRESENT";

  const eflhTable = {
    LOW_RISE: {
      PREWAR: 974,
      PRE79: 738,
      "1979_2006": 705,
      "2007PRESENT": 491
    },
    HIGH_RISE: {
      PREWAR: 987,
      PRE79: 513,
      "1979_2006": 385,
      "2007PRESENT": 214
    }
  };

  return eflhTable[buildingType][constructionEra];
}
```

The EFLH table categorizes buildings by:

- **Building Type**: Low-rise ($\leq$ 6 floors) vs High-rise ($>$ 6 floors)
- **Construction Era**: Pre-war ($\leq$ 1939), Pre-79 (1940-1978), 1979-2006, 2007-Present
- **EFLH Range**: 214-987 hours depending on building characteristics

3.1.3 Step 3: Convert kWh to MMBtu

Finally, convert the kWh result to MMBtu for consistency with other calculations:

$$\texttt{annualBuildingMMBtuHeatingPTHP} = \texttt{annualBuildingkWhHeatingPTHP} \times 0.003412 \tag{5}$$

Where 0.003412 is the conversion factor from kWh to MMBtu.

3.2 Cooling Energy

The cooling energy for PTHP remains equivalent to PTAC:

$$\texttt{annualBuildingMMBtuCoolingPTHP} = \texttt{annualBuildingMMBtuCoolingPTAC} \tag{6}$$

3.3 Total PTHP Energy

The total annual building energy consumption for PTHP:

$$\texttt{annualBuildingMMBtuTotalPTHP} = \texttt{annualBuildingMMBtuCoolingPTHP} + \texttt{annualBuildingMMBtuHeatingPTHP} \tag{7}$$

3.4 PTHP Building Totals for Cost Calculations

$$\texttt{annualBuildingKwhHeatingPTHP} = N_{ptacUnits} \times \texttt{annualUnitKwhHeatingPTHP} \tag{8}$$

$$\texttt{annualBuildingKwhCoolingPTHP} = N_{ptacUnits} \times \texttt{annualUnitKwhCoolingPTHP} \tag{9}$$

4 Energy Reduction Analysis

The energy reduction achieved by switching from PTAC to PTHP is calculated as:

$$\text{Reduction (\%)} = \frac{\texttt{annualBuildingMMBtuTotalPTAC} - \texttt{annualBuildingMMBtuTotalPTHP}}{\texttt{annualBuildingMMBtuTotalPTAC}} \times 100\% \tag{10}$$

5 Total Retrofit Cost Calculation

The total cost for retrofitting from PTAC to PTHP systems includes unit costs, installation costs, and contingency:

$$\texttt{totalRetrofitCost} = (\texttt{pthpUnitCost} + \texttt{pthpInstallationCost}) \times \text{Total PTHP Units} \times (1 + \texttt{pthpContingency}) \tag{11}$$

Example calculation:

$$\texttt{totalRetrofitCost} = (\$1,100 + \$450) \times N_{ptacUnits} \times 1.10 = \$1,705 \times N_{ptacUnits} \tag{12}$$

Where:

- pthpUnitCost: \$1,100 per PTHP unit
- pthpInstallationCost: \$450 per unit
- pthpContingency: 10% (1.10 multiplier)
- $N_{ptacUnits}$: Total number of PTHP units to be installed

6 Energy Cost Savings Calculation

The building energy totals for PTAC need to be expressed in kWh and therms so we can calculate price:

$$\texttt{annualBuildingThermsHeatingPTAC} = N_{ptacUnits} \times \texttt{annualUnitThermsHeatingPTAC} \tag{13}$$

$$\texttt{annualBuildingKwhCoolingPTAC} = N_{ptacUnits} \times \texttt{annualUnitKwhCoolingPTAC} \tag{14}$$

The annual energy cost savings from switching to PTHP systems is calculated as:

$$\begin{aligned}\texttt{annualBuildingCostPTAC} = {} & \texttt{annualBuildingKwhCoolingPTAC} \times \texttt{priceKwhHour} \\ & + \texttt{annualBuildingThermsHeatingPTAC} \times \texttt{priceThermHour}\end{aligned} \tag{15}$$

$$\begin{aligned}\texttt{annualBuildingCostPTHP} = {} & \texttt{annualBuildingKwhHeatingPTHP} \times \texttt{priceKwhHour} \\ & + \texttt{annualBuildingKwhCoolingPTHP} \times \texttt{priceKwhHour}\end{aligned} \tag{16}$$

$$\texttt{annualSavingsEnergy} = \texttt{annualBuildingCostPTAC} - \texttt{annualBuildingCostPTHP} \tag{17}$$

Where:

- priceKwhHour: $0.24 per kilowatt-hour for electricity (NYC average)
- priceThermHour: $1.45 per therm for natural gas (NYC average)[1]

7 LL97 Emissions Savings and BE Credit

7.1 Current Building Emissions and Budget

The building's current emissions are obtained from LL84 reporting:

$$\texttt{totalBuildingEmissionsLL84} = \text{Current building emissions (metric tons CO2e)} \tag{18}$$

The emissions budget for the building is calculated as the sum of type-specific emissions limits applied to corresponding square footage[2]:

$$\texttt{emissionsBudget} = \sum_i \texttt{squareFootageByType}_i \times \texttt{emissionsLimit}_i \tag{19}$$

7.2 Current Fee Calculation

LL97 has two compliance periods with different emissions budgets. The annual fee for exceeding the emissions budget is calculated separately for each period:

[1] Heating oil is priced at $2.60-$2.77 per therm, but most buildings use natural gas.

[2] Key LL97 property type limits (tCO2e/sf): Office (2024-29: 0.00846, 2030-34: 0.00453), Retail Store (2024-29: 0.01582, 2030-34: 0.00775), Multifamily Housing (2024-29: 0.00892, 2030-34: 0.00453), Hotel (2024-29: 0.01344, 2030-34: 0.00775), Hospital (2024-29: 0.02551, 2030-34: 0.01542), K-12 School (2024-29: 0.00846, 2030-34: 0.00453), Warehouse (2024-29: 0.00404, 2030-34: 0.00297). Complete mapping available in codebase at src/ll97/ll97_espm_to_bc_caps.json.

7.2.1 2024-2029 Compliance Period

$$\texttt{emissionsBudget2024to2029} = \sum_i \texttt{squareFootageByType}_i \times \texttt{emissionsLimit2024to2029}_i \tag{20}$$

$$\begin{aligned}\texttt{annualFeeExceedingBudget2024to2029} = (&\texttt{totalBuildingEmissionsLL84} \\ &- \texttt{emissionsBudget2024to2029}) \times \texttt{feePerTonCO2e}\end{aligned} \tag{21}$$

7.2.2 2030-2034 Compliance Period

$$\texttt{emissionsBudget2030to2034} = \sum_i \texttt{squareFootageByType}_i \times \texttt{emissionsLimit2030to2034}_i \tag{22}$$

$$\begin{aligned}\texttt{annualFeeExceedingBudget2030to2034} = (&\texttt{totalBuildingEmissionsLL84} \\ &- \texttt{emissionsBudget2030to2034}) \times \texttt{feePerTonCO2e}\end{aligned} \tag{23}$$

Where $\texttt{feePerTonCO2e} = \$268$ per tCO2e over budget, and the 2030-2034 period has more stringent (lower) emissions limits than the 2024-2029 period.

7.2.3 2034+ Compliance Period

$$\texttt{emissionsBudget2034toFuture} = \sum_i \texttt{squareFootageByType}_i \times \texttt{emissionsLimit2034toFuture}_i \tag{24}$$

$$\begin{aligned}\texttt{annualFeeExceedingBudget2034toFuture} = (&\texttt{totalBuildingEmissionsLL84} \\ &- \texttt{emissionsBudget2034toFuture}) \times \texttt{feePerTonCO2e}\end{aligned} \tag{25}$$

For financial modeling purposes, we assume:

$$\texttt{annualFeeExceedingBudget2034toFuture} = \texttt{annualFeeExceedingBudget2030to2034} \tag{26}$$

This assumption is necessary for downstream financial calculations[3].

7.3 BE Credit Section (Beneficial Electrification)

7.3.1 BE Credit Calculation

The BE credit is calculated based on heating electrification only, using annual heating kWh and time-dependent coefficients:

$$\texttt{beCreditBefore2027} = \texttt{annualBuildingkWhHeatingPTHP} \times 0.0013 \tag{27}$$

$$\texttt{beCredit2027to2029} = \texttt{annualBuildingkWhHeatingPTHP} \times 0.00065 \tag{28}$$

Where:

- Before January 1, 2027: Coefficient = 0.0013 tCO e/kWh
- 2027-2029: Coefficient = 0.00065 tCO e/kWh

[3] Post-2034 emissions budgets and compliance requirements are to be determined by NYC regulations. This conservative assumption maintains continuity for financial modeling purposes.

7.3.2 Adjusted Building Emissions Calculation

To accurately assess the LL97 fee impact of the PTAC→PTHP retrofit, we must calculate the building's adjusted emissions that reflect the actual post-retrofit energy consumption and emissions profile, rather than using the baseline LL84 emissions[4].

The adjusted building emissions are calculated as:

$$\begin{aligned}\texttt{adjustedTotalBuildingEmissions} = \texttt{totalBuildingEmissionsLL84} \\ - (\texttt{annualBuildingMMBtuHeatingPTAC} \times \texttt{efGas}) \\ + (\texttt{annualBuildingKwhHeatingPTHP} \times \texttt{efGrid\_kWh(year)})\end{aligned} \tag{29}$$

Where the emissions factors are:

- $\texttt{efGas}$: 0.05311 tCO e/MMBtu (natural gas factor, constant 2024-2034)
- $\texttt{efGrid\_kWh(year)}$: Grid electricity factor (tCO e/kWh) varying by compliance period:
  - $\texttt{efGrid2024to2029}$: 0.000288962 tCO e/kWh
  - $\texttt{efGrid2030to2034}$: 0.000145 tCO e/kWh

This gives us period-specific adjusted emissions:

$$\begin{aligned}\texttt{adjustedTotalBuildingEmissions2024to2029} = \texttt{totalBuildingEmissionsLL84} \\ - (\texttt{annualBuildingMMBtuHeatingPTAC} \times 0.05311) \\ + (\texttt{annualBuildingKwhHeatingPTHP} \times 0.000288962)\end{aligned} \tag{30}$$

$$\begin{aligned}\texttt{adjustedTotalBuildingEmissions2030to2034} = \texttt{totalBuildingEmissionsLL84} \\ - (\texttt{annualBuildingMMBtuHeatingPTAC} \times 0.05311) \\ + (\texttt{annualBuildingKwhHeatingPTHP} \times 0.000145)\end{aligned} \tag{31}$$

7.3.3 Annual Fee with BE Credit (Before 2027)

For buildings upgrading before January 1, 2027:

$$\texttt{newEmissionsBefore2027} = \texttt{adjustedTotalBuildingEmissions2024to2029} - \texttt{beCreditBefore2027} \tag{32}$$

$$\begin{aligned}\texttt{adjustedAnnualFeeBefore2027} = (\texttt{newEmissionsBefore2027} \\ - \texttt{emissionsBudget2024to2029}) \times \texttt{feePerTonCO2e}\end{aligned} \tag{33}$$

7.3.4 Annual Fee with BE Credit (2027-2029)

For buildings upgrading between 2027-2029:

$$\texttt{newEmissions2027to2029} = \texttt{adjustedTotalBuildingEmissions2024to2029} - \texttt{beCredit2027to2029} \tag{34}$$

$$\begin{aligned}\texttt{adjustedAnnualFee2027to2029} = (\texttt{newEmissions2027to2029} \\ - \texttt{emissionsBudget2024to2029}) \times \texttt{feePerTonCO2e}\end{aligned} \tag{35}$$

[4] Using raw LL84 emissions would not account for the actual emissions reduction from switching from gas heating to electric heat pumps, which varies by time period due to changing grid electricity emissions factors.

7.3.5 Annual Fee for 2030-2034

For the 2030-2034 period, there is no BE credit available, so the fee is calculated using the adjusted emissions for this period:

$$\begin{aligned}\texttt{adjustedAnnualFee2030to2034} = (&\texttt{adjustedTotalBuildingEmissions2030to2034} \\ &- \texttt{emissionsBudget2030to2034}) \times \texttt{feePerTonCO2e}\end{aligned} \tag{36}$$

7.3.6 Annual Fee for 2034+

For the post-2034 period, there is no BE credit available, so the fee is calculated using the adjusted emissions for this period:

$$\begin{aligned}\texttt{adjustedTotalBuildingEmissions2034toFuture} = &\ \texttt{totalBuildingEmissionsLL84} \\ &- (\texttt{annualBuildingMMBtuHeatingPTAC} \times 0.05311) \\ &+ (\texttt{annualBuildingKwhHeatingPTHP} \times 0.000145)\end{aligned} \tag{37}$$

$$\begin{aligned}\texttt{adjustedAnnualFee2034toFuture} = (&\texttt{adjustedTotalBuildingEmissions2034toFuture} \\ &- \texttt{emissionsBudget2034toFuture}) \times \texttt{feePerTonCO2e}\end{aligned} \tag{38}$$

For modeling purposes, we assume adjustedTotalBuildingEmissions2034toFuture uses the same grid electricity emissions factor as the 2030-2034 period[5].

8 Financial Analysis of PTHP Upgrade

In order to calculate the financial viability of upgrading from PTAC to PTHP systems, we first need to calculate how much money you will be saving each year from the time you upgrade. This annual savings is composed of two main components: the energy savings (annualSavingsEnergy from Section 6) and the LL97 fee avoidance. The following 4 subsections show how calculating this varies depending on which period you are in, noting that LL97 compliance continues post-2034 with ongoing fee avoidance benefits.

8.1 LL97 Fee Avoidance Calculations

8.1.1 LL97 Fee Avoidance for 2024-2027

$$\begin{aligned}\texttt{annualLL97FeeAvoidance2024to2027} = &\ \texttt{annualFeeExceedingBudget2024to2029} \\ &- \texttt{adjustedAnnualFeeBefore2027}\end{aligned} \tag{39}$$

8.1.2 LL97 Fee Avoidance for 2027-2029

$$\begin{aligned}\texttt{annualLL97FeeAvoidance2027to2029} = &\ \texttt{annualFeeExceedingBudget2024to2029} \\ &- \texttt{adjustedAnnualFee2027to2029}\end{aligned} \tag{40}$$

8.1.3 LL97 Fee Avoidance for 2030-2034

$$\begin{aligned}\texttt{annualLL97FeeAvoidance2030to2034} = &\ \texttt{annualFeeExceedingBudget2030to2034} \\ &- \texttt{adjustedAnnualFee2030to2034}\end{aligned} \tag{41}$$

[5] Future grid electricity emissions factors post-2034 are not yet determined. This assumption provides continuity for financial modeling while maintaining the conservative approach of using known emissions factors.

8.1.4 LL97 Fee Avoidance for 2034+

$$\begin{aligned}\texttt{annualLL97FeeAvoidance2034toFuture} &= \texttt{annualFeeExceedingBudget2034toFuture} \\ &\quad - \texttt{adjustedAnnualFee2034toFuture} \qquad (42)\end{aligned}$$

8.2 Code Implementation

This is the implementation for calculating cumulative savings:

Code Chunk 1: Function to calculate annual savings in a given year

```
function getAnnualSavings(year: number): number {
  const energySavings = annualSavingsEnergy;

  if (year >= 2024 && year <= 2026) {
    return energySavings + annualLL97FeeAvoidance2024to2027;
  } else if (year >= 2027 && year <= 2029) {
    return energySavings + annualLL97FeeAvoidance2027to2029;
  } else if (year >= 2030 && year <= 2034) {
    return energySavings + annualLL97FeeAvoidance2030to2034;
  } else {
    return energySavings + annualLL97FeeAvoidance2034toFuture;
  }
}
```

Code Chunk 2: Example usage of calculating annual savings

```
// Loan term years for a static example - upgrade completes in 2025, savings start in 2026
const loanYears = [2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037,
    2038, 2039, 2040];
const cumulativeSavingsByYear = [];
let cumulativeSavings = 0;

for (const year of loanYears) {
  // No savings in 2025 (upgrade year), savings begin in 2026
  const annualSavings = year === 2025 ? 0 : getAnnualSavings(year);
  cumulativeSavings += annualSavings;

  cumulativeSavingsByYear.push(cumulativeSavings);
}

// This array will be the data for the green cumulative savings line in the visualization
console.log(cumulativeSavingsByYear);

// Actual output: [0, 47000, 94000, 138000, 182000, 226000, 267000, 308000, 349000, 390000,
    431000, 468000, 505000, 542000, 579000, 616000]
```

From the array of cumulative savings year-by-year, once it gets above the totalRetrofitCost, then that year is the Simple Payback Period.[6]

8.3 Payback Period Implementation

For `totalRetrofitCost` = $500,000, the cumulative savings array shows:

[0, 47000, 94000, 138000, 182000, 226000, 267000, 308000, 349000, 390000, 431000, 468000, **505000**, 542000, 579000, 616000]

Year 2037 achieves payback with $505,000 cumulative savings.

9 Loan Financing and Payback Visualization

This section demonstrates how loan financing affects the payback period through a visual representation of loan balance versus cumulative savings over time.

[6] Simple division (totalRetrofitCost ÷ averageAnnualSavings) cannot be used here because annual savings vary significantly across LL97 compliance periods. The year-by-year calculation accounts for these varying savings rates.

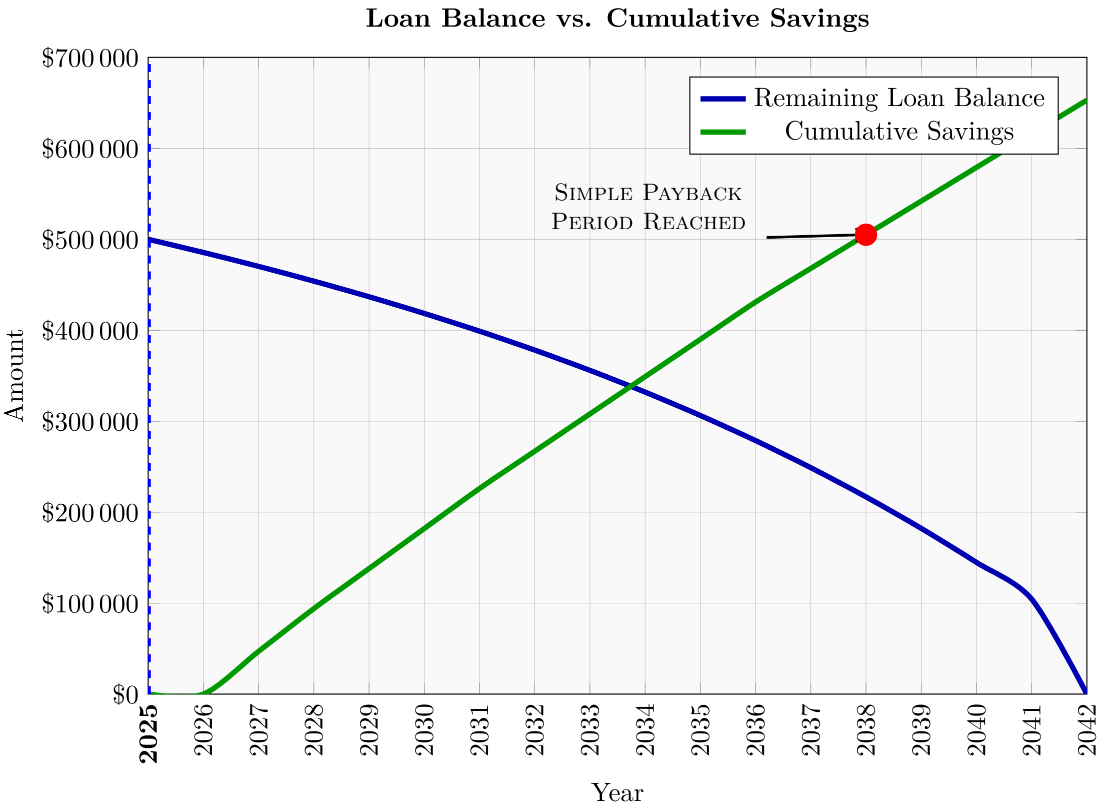
9.1 Loan Parameters

For this example, we use typical commercial loan parameters[7]:

The loan calculations use standard amortization formulas[8] and the remaining balance formula[9].

Cumulative savings grow linearly over time[10]:

9.2 Visualization



The simple payback period occurs when the green cumulative savings line reaches the loan principal amount of $500,000. This visualization shows how cumulative savings continue to grow beyond payback, demonstrating the long-term financial benefits of the retrofit investment.

10 NOI Analysis

Net Operating Income (NOI) is obtained from external building data APIs[11] and represents the baseline building income before upgrade improvements.

10.1 Upgrade Impact on NOI

The true impact of PTHP upgrades becomes clear when comparing NOI scenarios:

[7] Example parameters: Principal = $500,000 (totalRetrofitCost), Term = 15 years, Annual interest rate = 6%, Monthly interest rate = 0.005, Total payments = 180.

[8] Monthly payment: $P \times \frac{r(1+r)^n}{(1+r)^n-1}$ where P=principal, r=monthly rate, n=total payments.

[9] Remaining balance: $P \times \frac{(1+r)^n-(1+r)^{mt}}{(1+r)^n-1}$ where t=years elapsed, m=12 months/year.

[10] Simplified as: Cumulative Savings(t) = t × Average Annual Savings. In reality, LL97 fee avoidance varies by compliance period.

[11] NOI data sources: Co-ops/Condominiums use direct NYC Open Data API (net_operating_income); Rental Buildings calculate from rentable income: Gross Income = (Building Sq Ft × 0.8 × $6/sq ft × 12 months) × 1.108; NOI = Gross Income × (1 - 0.55) for OpEx + taxes.

Scenario A: No Upgrade (Status Quo)

$$\texttt{noiNoUpgrade2024to2029} = \texttt{currentNOI} - \texttt{annualFeeExceedingBudget2024to2029} \tag{43}$$

$$\texttt{noiNoUpgrade2030to2034} = \texttt{currentNOI} - \texttt{annualFeeExceedingBudget2030to2034} \tag{44}$$

$$\texttt{noiNoUpgradePost2034} = \texttt{currentNOI} - \texttt{annualFeeExceedingBudget2034toFuture} \tag{45}$$

Scenario B: With PTHP Upgrade

$$\begin{aligned}\texttt{noiWithUpgrade2024to2027} = \texttt{currentNOI} &+ \texttt{annualSavingsEnergy}\\ &- \texttt{adjustedAnnualFee2024to2027}\end{aligned} \tag{46}$$

$$\begin{aligned}\texttt{noiWithUpgrade2027to2029} = \texttt{currentNOI} &+ \texttt{annualSavingsEnergy}\\ &- \texttt{adjustedAnnualFee2027to2029}\end{aligned} \tag{47}$$

$$\begin{aligned}\texttt{noiWithUpgrade2030to2034} = \texttt{currentNOI} &+ \texttt{annualSavingsEnergy}\\ &- \texttt{adjustedAnnualFee2030to2034}\end{aligned} \tag{48}$$

$$\begin{aligned}\texttt{noiWithUpgradePost2034} = \texttt{currentNOI} &+ \texttt{annualSavingsEnergy}\\ &- \texttt{adjustedAnnualFee2034toFuture}\end{aligned} \tag{49}$$

10.2 NOI Calculation Functions

The NOI calculations can be implemented with unified functions that handle all compliance periods:

```
function calculateAdjustedNOINoUpgrade(year: number): number {
  const currentNOI = getCurrentNOI(); // from external API

  if (year >= 2024 && year <= 2029) {
    return currentNOI - annualFeeExceedingBudget2024to2029;
  } else if (year >= 2030 && year <= 2034) {
    return currentNOI - annualFeeExceedingBudget2030to2034;
  } else {
    return currentNOI - annualFeeExceedingBudget2034toFuture;
  }
}

function calculateAdjustedNOIUpgrade(year: number): number {
  const currentNOI = getCurrentNOI(); // from external API
  const energySavings = getEnergySavings();

  let reducedLL97Penalties = 0;
  if (year >= 2024 && year <= 2029) {
    reducedLL97Penalties = adjustedAnnualFee2024to2029; // actual penalties paid after upgrade
  } else if (year >= 2030 && year <= 2034) {
    reducedLL97Penalties = adjustedAnnualFee2030to2034; // actual penalties paid after upgrade
  } else {
    reducedLL97Penalties = adjustedAnnualFee2034toFuture; // actual penalties paid after upgrade
  }

  return currentNOI + energySavings - reducedLL97Penalties;
}
```

10.2.1 Implementation Example

For a 100,000 sq ft rental building with current NOI of $1,200,000:

Without Upgrade: Building faces ongoing LL97 penalties starting in 2026

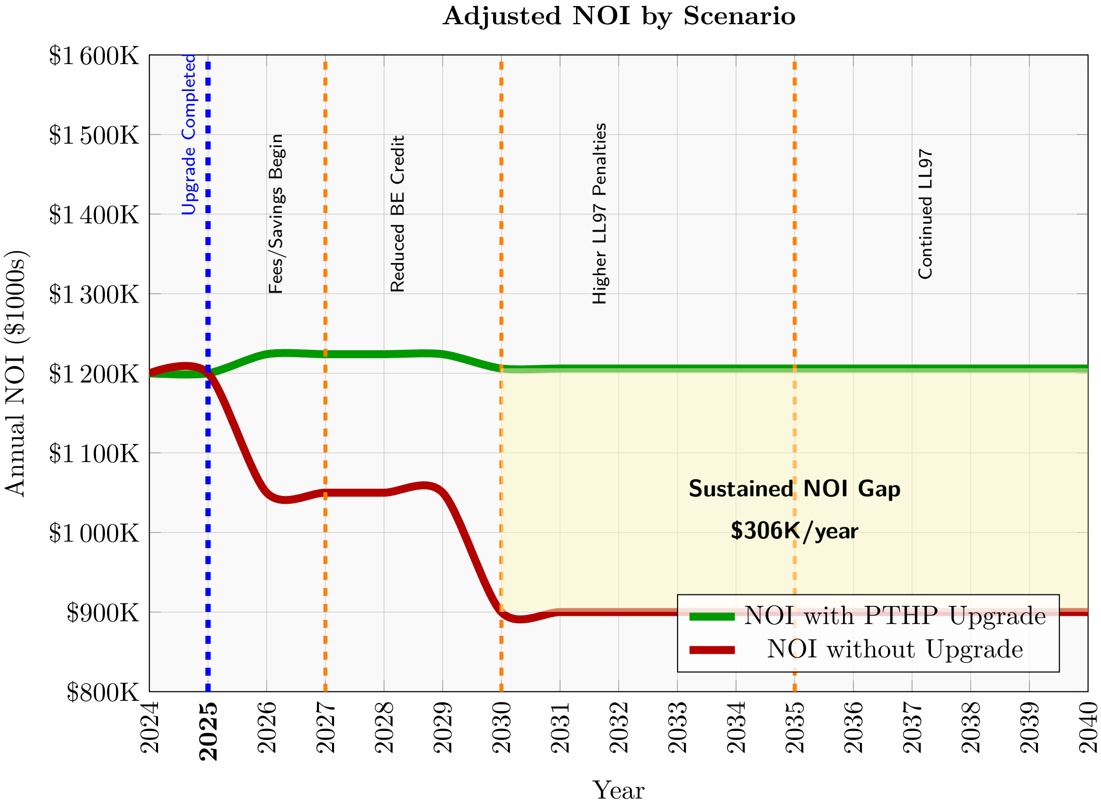
- 2024-2025: NOI = $1,200,000 (current NOI) - $0 (no penalties yet) = $1,200,000
- 2026-2029: NOI = $1,200,000 (current NOI) - $150,000 (LL97 penalties paid) = $1,050,000
- 2030-2034: NOI = $1,200,000 (current NOI) - $300,000 (LL97 penalties paid) = $900,000
- 2035+: NOI = $1,200,000 (current NOI) - $300,000 (LL97 penalties paid) = $900,000

With PTHP Upgrade: Upgrade completes in 2025, savings + reduced penalties start in 2026

- 2024-2025: NOI = $1,200,000 (current NOI) - $0 (no changes yet) = $1,200,000
- 2026-2027: NOI = $1,200,000 (current NOI) + $39,000 (energy savings) - $15,000 (reduced LL97 penalties) = $1,224,000
- 2030-2034: NOI = $1,200,000 (current NOI) + $39,000 (energy savings) - $33,000 (reduced LL97 penalties) = $1,206,000
- 2035+: NOI = $1,200,000 (current NOI) + $39,000 (energy savings) - $33,000 (reduced LL97 penalties) = $1,206,000

10.3 NOI Visualization Over Time

The following chart shows how NOI evolves differently in upgrade vs no-upgrade scenarios across LL97 compliance periods:



This NOI comparison reveals the true financial impact of upgrade timing:

- **Immediate NOI Boost**: Upgrading in 2024 increases NOI by $174K/year (17% increase)
- **Penalty Avoidance**: Without upgrade, NOI drops $150K in 2030+ due to higher penalties
- **Perpetual Benefit**: The $306K annual NOI gap continues indefinitely post-2034, creating sustained property value advantages
- **Risk Mitigation**: Upgrading protects against declining property values from ongoing LL97 penalties
- **Optimal Window**: Early upgrade (2024-2027) captures maximum BE Credit and avoids penalty escalation
- **Long-term Value**: LL97 compliance continues post-2034, making upgrade benefits permanent rather than temporary

11 Property Value Analysis

Property values are calculated by dividing NOI by the cap rate, providing a direct correlation between operational performance improvements and asset valuation.

11.1 Property Value Calculation Functions

Property value calculations use the same scenarios as NOI but apply cap rate conversion:

```
function calculatePropertyValueNoUpgrade(year: number, capRate: number = 0.04): number {
  const adjustedNOI = calculateAdjustedNOINoUpgrade(year); // NOI minus actual LL97 penalties paid
  return adjustedNOI / capRate;
}

function calculatePropertyValueUpgrade(year: number, capRate: number = 0.04): number {
  const adjustedNOI = calculateAdjustedNOIUpgrade(year); // NOI plus energy savings minus reduced LL97 penalties
  return adjustedNOI / capRate;
}
```

The default cap rate of 4% reflects typical NYC multifamily properties[12].

11.2 Property Value Impact Analysis

Using a 4% cap rate typical for NYC multifamily properties:

No Upgrade Property Values

$$\texttt{propertyValueNoUpgrade} = \texttt{noiNoUpgrade} \div 0.04 \tag{50}$$

With Upgrade Property Values

$$\texttt{propertyValueWithUpgrade} = \texttt{noiWithUpgrade} \div 0.04 \tag{51}$$

Net Property Value Impact

$$\texttt{netPropertyValueGain} = \texttt{propertyValueWithUpgrade} - \texttt{propertyValueNoUpgrade} \tag{52}$$

[12] Cap rate of 4% used for NYC multifamily properties. This parameter can be adjusted based on market conditions, property characteristics, and investment criteria.

11.2.1 Implementation Example

For a 100,000 sq ft rental building with current NOI of $1,200,000:

Without Upgrade: Building faces ongoing LL97 penalties starting in 2026

- 2024-2025: Property Value = $1,200,000 (NOI) ÷ 0.04 (cap rate) = $30.0M
- 2026-2029: Property Value = $1,050,000 (NOI) ÷ 0.04 (cap rate) = $26.25M
- 2030-2034: Property Value = $900,000 (NOI) ÷ 0.04 (cap rate) = $22.5M
- 2035+: Property Value = $900,000 (NOI) ÷ 0.04 (cap rate) = $22.5M

With PTHP Upgrade: Upgrade completes in 2025, value increases in 2026

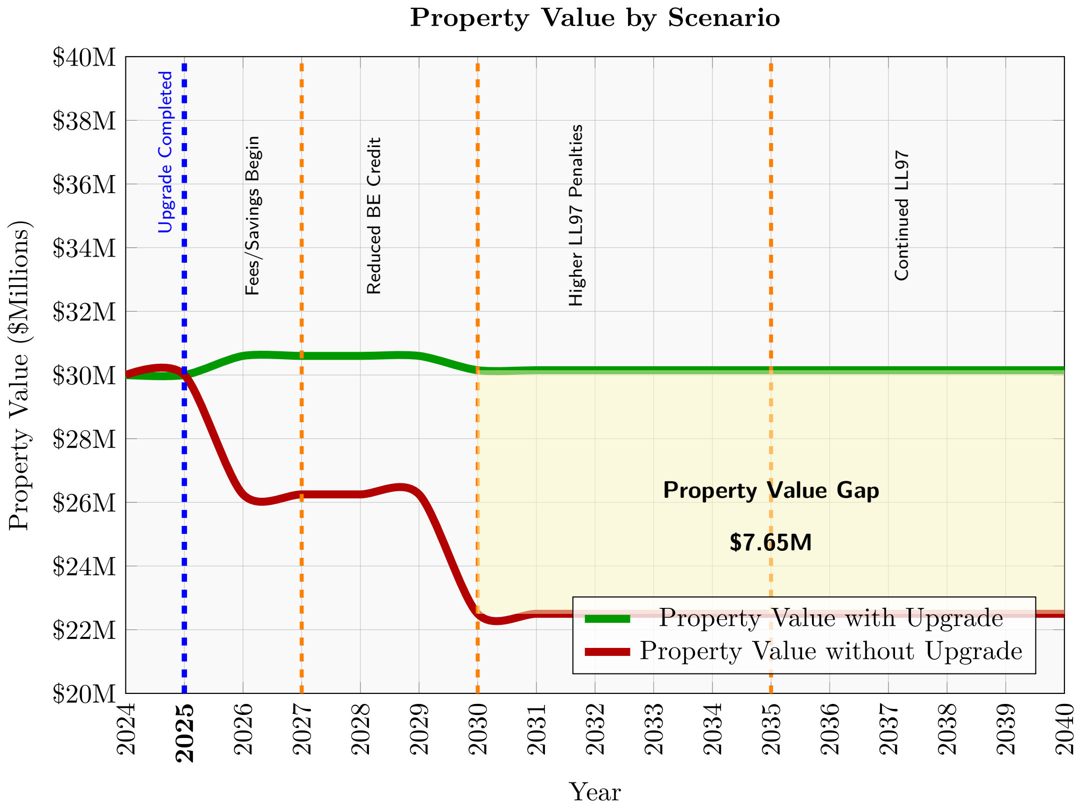
- 2024-2025: Property Value = $1,200,000 (NOI) ÷ 0.04 (cap rate) = $30.0M
- 2026-2027: Property Value = $1,224,000 (NOI) ÷ 0.04 (cap rate) = $30.6M
- 2030-2034: Property Value = $1,206,000 (NOI) ÷ 0.04 (cap rate) = $30.15M
- 2035+: Property Value = $1,206,000 (NOI) ÷ 0.04 (cap rate) = $30.15M

Net Property Value Gain: $30.15M - $22.5M = $7.65M

This $7.65M property value increase far exceeds typical retrofit costs of $500K-$2M, demonstrating the compelling financial case for upgrading.

11.3 Property Value Impact Visualization

The following chart demonstrates property value trajectories for both scenarios:



This visualization demonstrates the substantial and sustained property value advantage of upgrading, with the $7.65M gap representing the permanent value creation from PTHP installation.

12 Implementation Notes

All variables in the codebase follow camelCase naming convention with the appropriate PTAC or PTHP suffix for clarity. The calculations are implemented in the backend services using TypeScript, ensuring type safety and consistent naming across the platform.