

FONDEMENTS DU PROJET.

“ Le besoin le plus important aujourd'hui pour développer la paille en zone urbaine est de former plus d'acteurs à ce type de construction. Il faut pouvoir transmettre ce savoir dans un centre dédié aux formations [...] et de pouvoir sensibiliser le public à ces questions. ”



Benoît Rougelot, Architecte et Co-président du Collect'if Paille.

“ L'Île-de-France peut mobiliser 234 945 tonnes de paille par an pour le bâtiment. Agence Régionale de l'Environnement et des Nouvelles Énergies. ”



C'est l'équivalent de 8223 maisons de 100m² ou de 300 ERP de 8615m²

“ Nous essayons de porter un projet de maison de l'habitat décentralisé ancré au territoire qui rassemblerait les ressources sur l'habitat durable et montrerait les nouvelles manières d'habiter. ”

L'enjeu majeur du neuf aujourd'hui, c'est le réemploi et la mutabilité des constructions

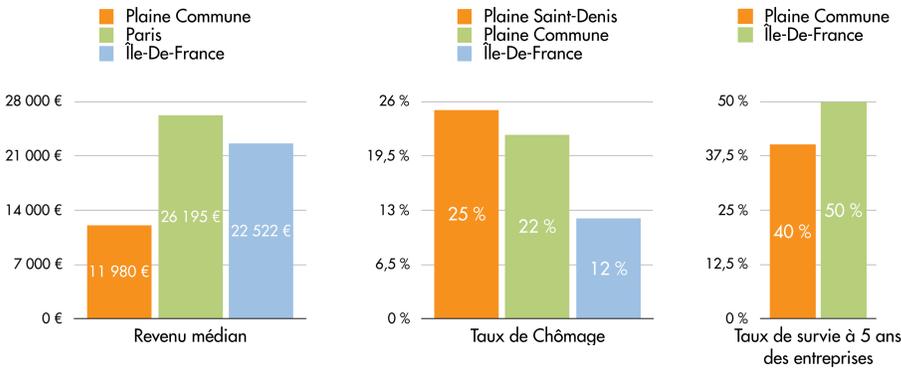


Michael Evrard,
Secrétaire général de l'Agence Local de l'Énergie et du Climat.

“ L'Île-de-France compte 260 000 hectares de forêt soit un taux de boisement de 21% proche de la moyenne nationale de 31%. La majorité du bois prélevé en Île-de-France est exporté. ”

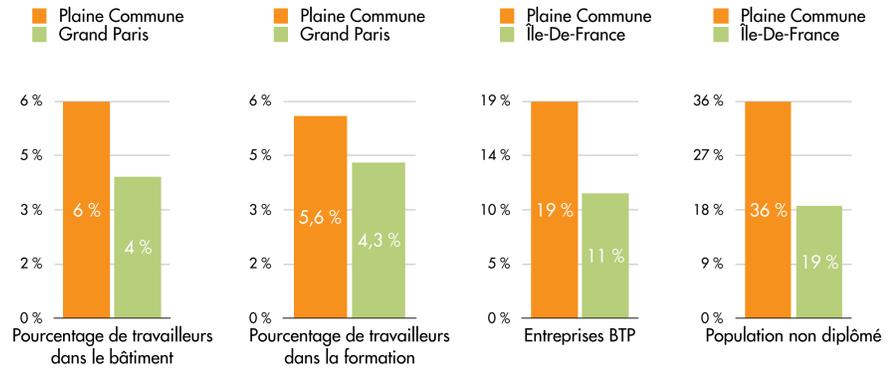


Plaine Saint-Denis : Un territoire à soutenir

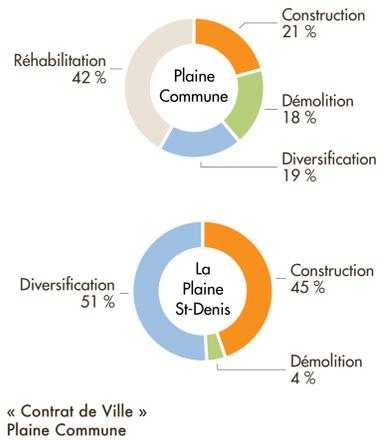


La Plaine Saint-Denis : Un potentiel spécifique

Source : Insee 2016, 2014, « Contrat de Ville » Plaine Commune 2015



La Plaine : Un Bâti renouvelé constamment



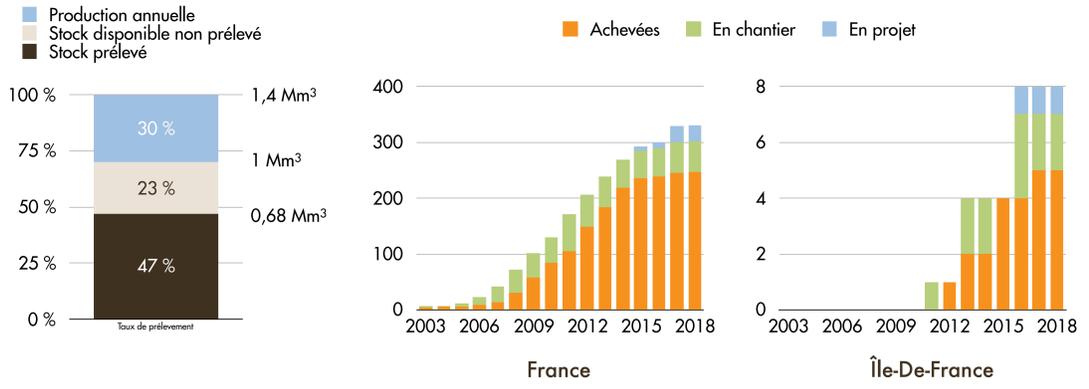
« Contrat de Ville » Plaine Commune

Des lieux de culture et d'échanges manquants



Source : « Le contrat de ville » - Préfecture, Plaine Commune, Grand Paris 2015

Les matériaux biosourcés, des avantages reconnus

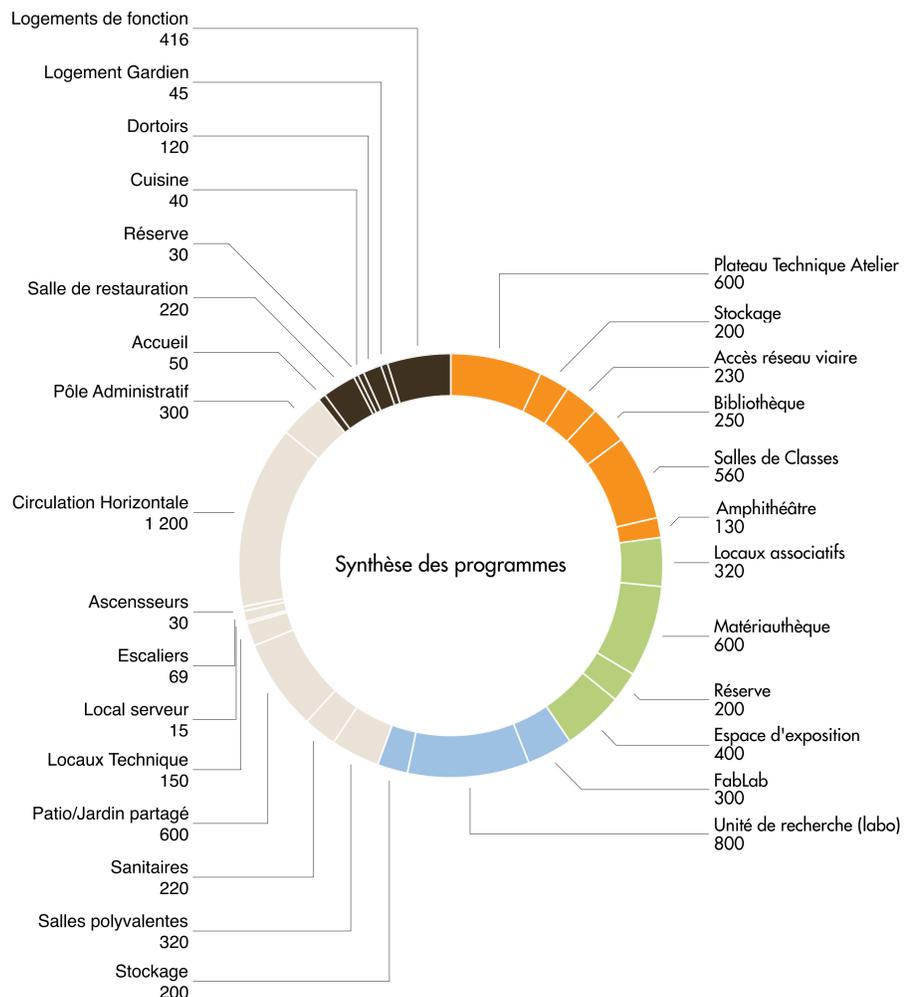


Le prélèvement de bois en IDF

Le nombre de constructions paille.

Source : Collect'if Paille d'Île-de-France

Points de données	Catégorie	Fonction	Espace exact	Nb	Surface/u m ²	Surface m ²
Plateau Technique Atelier	Formations	Plateau Technique Atelier	Plateau	1	600	600
Stockage			Stockage	1	200	200
Accès réseau viaire			Accès réseau viaire	1	230	230
Bibliothèque		Bibliothèque		1	250	250
Salles de Classes		Salles de Classes		8	70	560
Amphithéâtre		Amphithéâtre		1	130	130
Locaux associatifs	Sensibilisation	Associations	Locaux associatifs	4	80	320
Matériautheque		Matériautheque	Matériautheque	1	600	600
Réserve			Réserve	1	200	200
Espace d'exposition		Exposition	Espace d'exposition	1	400	400
FabLab	Recherche	FabLab		1	300	300
Unité de recherche (labo)		Unité de recherche (labo)		2	400	800
Stockage		Stockage		1	200	200
Salles polyvalentes	Mutualisé		Salles polyvalentes	4	80	320
Sanitaires			Sanitaires	4	55	220
Patio/Jardin partagé			Patio/Jardin partagé	2	300	600
Locaux Technique			Locaux Technique	3	50	150
Local serveur			Local serveur	1	15	15
Escaliers		Circulation verticale	Escaliers	3	23	69
Ascenseurs			Ascenseurs	3	10	30
Circulation Horizontale		Circulation Horizontale	Dessertes	4	300	1 200
Pôle Administratif		Pôle Administratif		1	300	300
Accueil	Général	Accueil		1	50	50
Salle de restauration		Complexe de restauration / Café	Salle de restauration	1	220	220
Réserve			Réserve	1	30	30
Cuisine			Cuisine	1	40	40
Dortoirs	Hébergement	Dortoirs		1	120	120
Logement Gardien		Logement Gardien		1	45	45
Logements de fonction		Logements de fonction		8	52	416
		Synthèse		63	184	8 615



PROJET DE CENTRE DE FORMATION, DE RECHERCHE ET DE SENSIBILISATION AUX MATÉRIAUX BIOSOURCÉS

ESSAI SUR L'ASSISTANCE NUMÉRIQUE POUR L'ORGANISATION SPATIALE

Script de saisie des données de comparaisons

```
# dataProgramsScore = [[0,0,0,0,0,0],[0,0,0,0,0,0],[0,0,0,0,0,0],[0,0,0,0,0,0]]
dataProgramsNames = ['Atelier', 'Amphi', 'Classes', 'Bibli', 'Expo', 'Mtheque', 'Assos', 'Admin', 'Accueil', 'Fablab', 'La-
bo', 'Patio', 'Logements', 'Resto']
dataProgramsScore = []
originValue = 0
fileName='outputfile.csv'

my_file = open(fileName, 'w')
my_file.write('Venez donc tester le MMR de votre programme.')
my_file.close()

# Créer un tableau à deux dimensions avec le score de lien de chaque programme avec chaque autre.
for s in range(len(dataProgramsNames)):
    originScoreList=[]
    for v in range(len(dataProgramsNames)):
        originScoreList.append(originValue)
    dataProgramsScore.append(originScoreList)

'''Listes pour test'''
dataProgramsNames=['Atelier', 'Amphi', 'Classes', 'Bibli']
dataProgramsScore=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]]

c=0#compteur de comparaisons reellement demandees

for boucle1 in range(len(dataProgramsNames)-1,-1,-1):
    print("Impression boucle1 - B1=%s"%(boucle1))
    memory=[]
    for boucle2 in range(len(dataProgramsNames)-1,-1,-1):
        print("Impression boucle2 - B1=%s, B2=%s"%(boucle1,boucle2))
        for boucle3 in range(len(dataProgramsNames)-1,-1,-1):
            print("Impression boucle3 - B1=%s, B2=%s, B3=%s"%(boucle1,boucle2,boucle3))

            choice = 'Undefined'
            itemBase=dataProgramsNames[boucle1]
            item1=dataProgramsNames[boucle2]
            item2=dataProgramsNames[boucle3]

            # Pour se souvenir de combinaisons deja examine sans cramer la memoire.
            binome=[]
            binome.append(boucle2)
            binome.append(boucle3)
            binome.sort()
            polynome=[]
            polynome.append(boucle1)
            polynome.append(binome[0])
            polynome.append(binome[1])
            # print("Le polynome : %s et boucle1 : %s, et binome : %s"%(polynome,boucle1,binome))

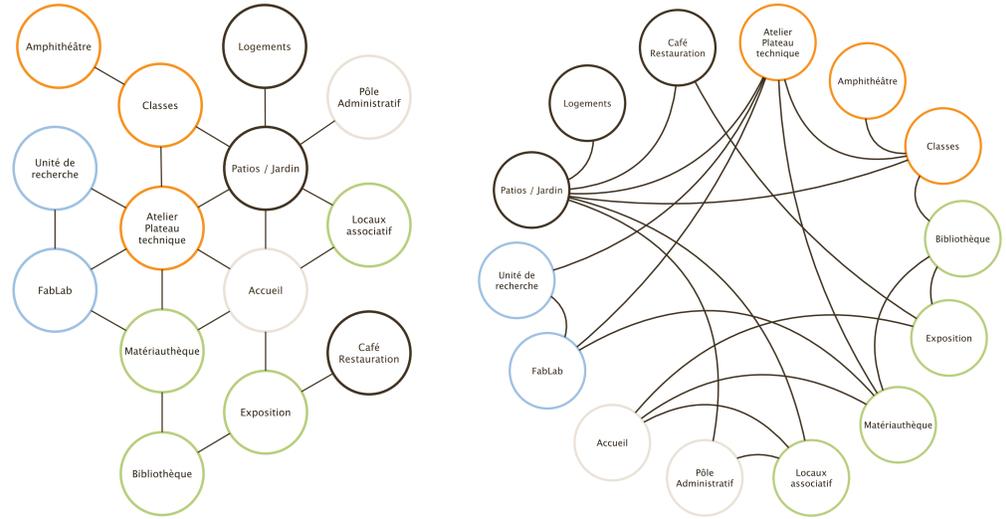
            if item1 == item2 or item1==itemBase or item2==itemBase or (polynome in memory):
                if memory and not (item1 == item2 or item1==itemBase or item2==itemBase):
                    print("Notre memoire nous a empeche de faire une betise : %s, %s, %s"%(itemBase,item1,item2))
                else :
                    print("On essaye de sauter une boucle : %s, %s, %s"%(itemBase,item1,item2))
                pass
            else :
                while choice!="1" or choice!="2":
                    choice = input("Le programme %s est-il plus lie au programme %s ou %s ?\n"%(itemBase,item1,item2))
                    if choice == "1":
                        dataProgramsScore[boucle1][boucle2]+=1
                        printCoor=boucle2
                        break
                    elif choice == "2":
                        dataProgramsScore[boucle1][boucle3]+=1
                        printCoor=boucle3
                        break
                    else:
                        print("Y'a une couille dans le potage ! Recommence pour cette paire :)")

            if (polynome not in memory) and (len(polynome)==3):
                memory.append(polynome)
            else:
                pass
            print(dataProgramsScore)
            print("J'incremente +1 coor(%s,%s). Tableau memoire : %s"%(str(boucle1+1),str(printCoor+1),memory))
            c+=1#incrementation compteur de comparaison reellement demande.

l=c
my_file = open(fileName, 'a')
my_file.write('Ajout des %s comparaisons des programmes lies a %s.\n'%(l,dataProgramsNames[boucle1]))
my_file.write(str(dataProgramsScore[boucle1])+"\n")
my_file.close()
l=0

print(dataProgramsScore)
for i in range(len(dataProgramsScore)):
    print(dataProgramsScore[i])
print("C'est termine !")
```

Diagrammes de relation des programmes. - Vision à la main.



Un script qui mesure auprès du concepteur la proximité / le lien de l'ensemble des programmes deux à deux.

“Le programme «Atelier» est-il plus lié au programme «Unité de recherche» ou «Matériautheque» ?”

“Le programme «FabLab» est-il plus lié au programme «Locaux Associatifs» ou «Amphithéâtre» ?”

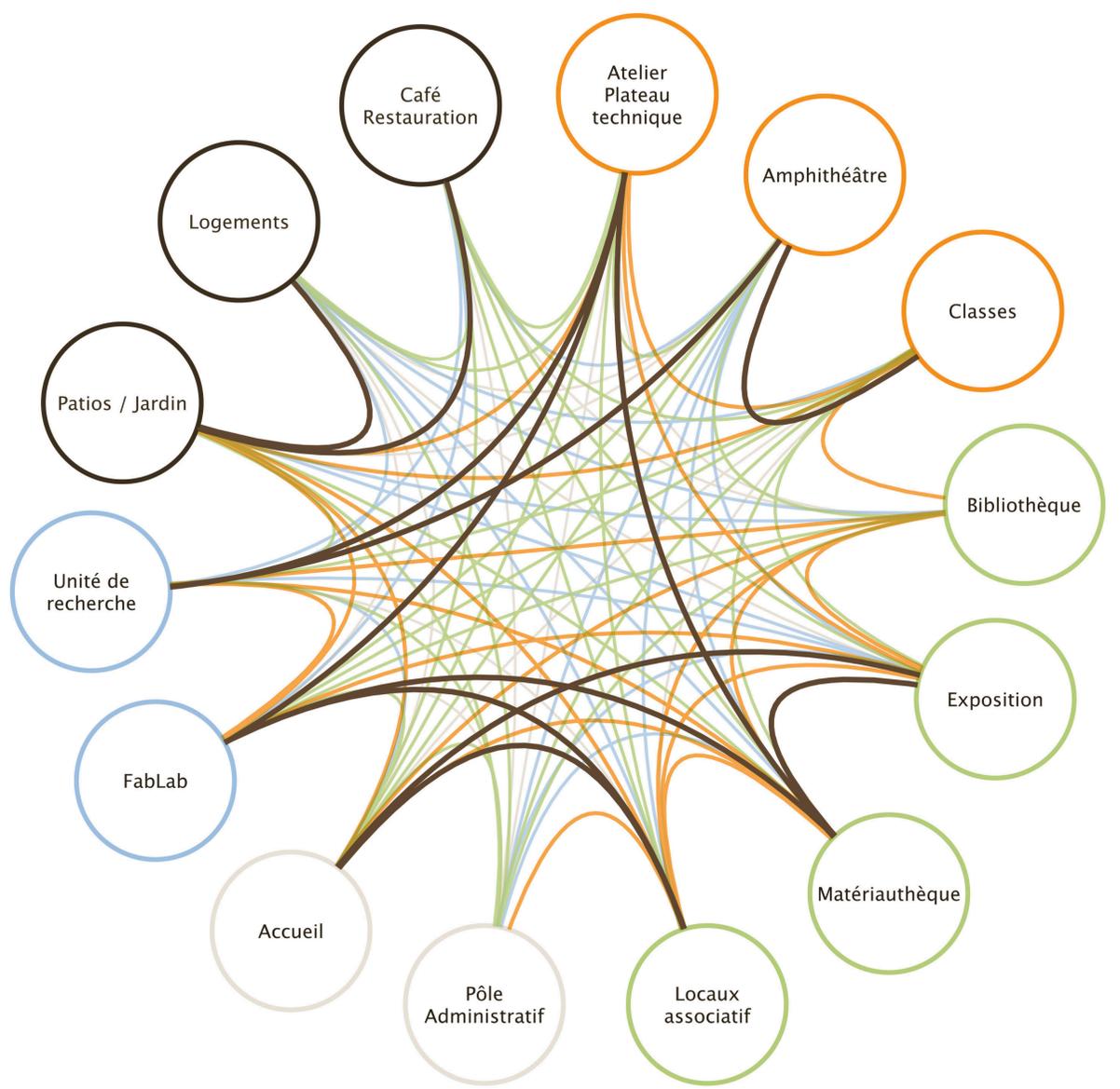
1092 Comparaisons

Score 20-24 (Black) Score 15-19 (Orange) Score 10-14 (Green) Score 5-9 (Blue) Score 0-4 (Grey)

Matrice des relations - 3 boucles (inception) non-orienté

Atelier	Amphi	Classes	Bibli	Expo	Mtheq	Assos	Admin	Accueil	Fablab	Labo	Patio	Logeme	Resto	m\n
	13	19	4	17	23	10	3	12	22	24	19	10	11	Atelier
		24	11	5	8	7	7	14	13	20	0	1	5	Amphi
			17	11	10	3	10	3	12	13	18	9	13	Classes
				15	17	15	13	15	12	16	6	6	2	Bibli
					21	15	8	23	15	6	9	2	11	Expo
						16	8	19	22	16	13	6	12	Mtheque
							17	24	21	2	15	8	13	Assos
								13	4	11	13	10	4	Admin
									12	5	15	3	10	Accueil
										17	15	4	8	Fablab
											10	6	9	Labo
												21	23	Patio
													11	Logements
														Resto

- Formation
- Sensibilisation
- Général
- Recherche
- Mutualisé
- Lien de force 5 Programmes fortement liés
- Lien de force 4
- Lien de force 3
- Lien de force 2
- Lien de force 1 Programmes peu liés



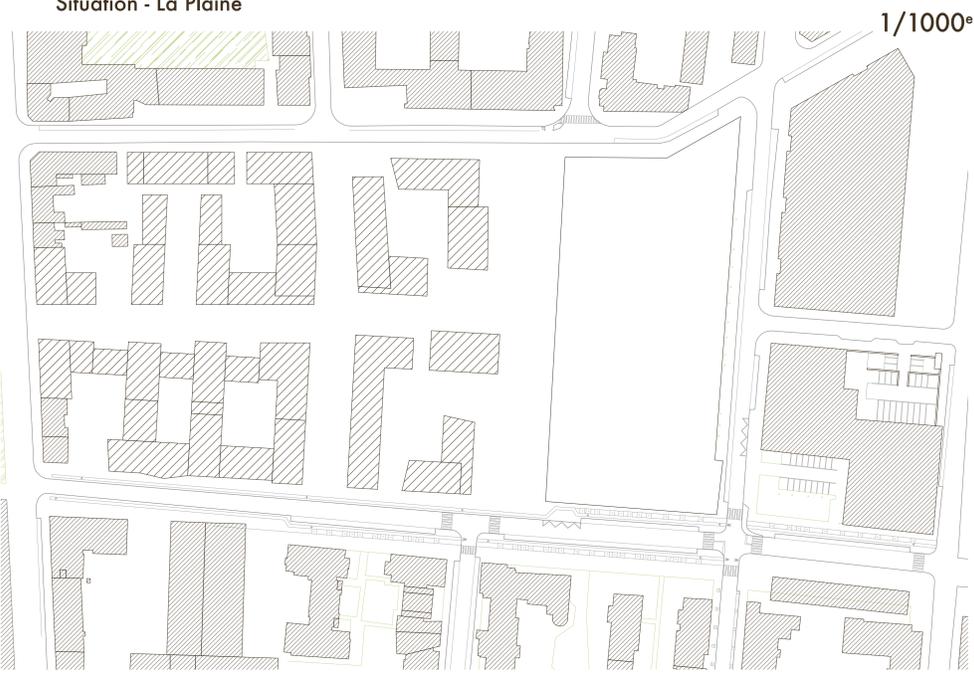
SITUATION



Situation - La Plaine

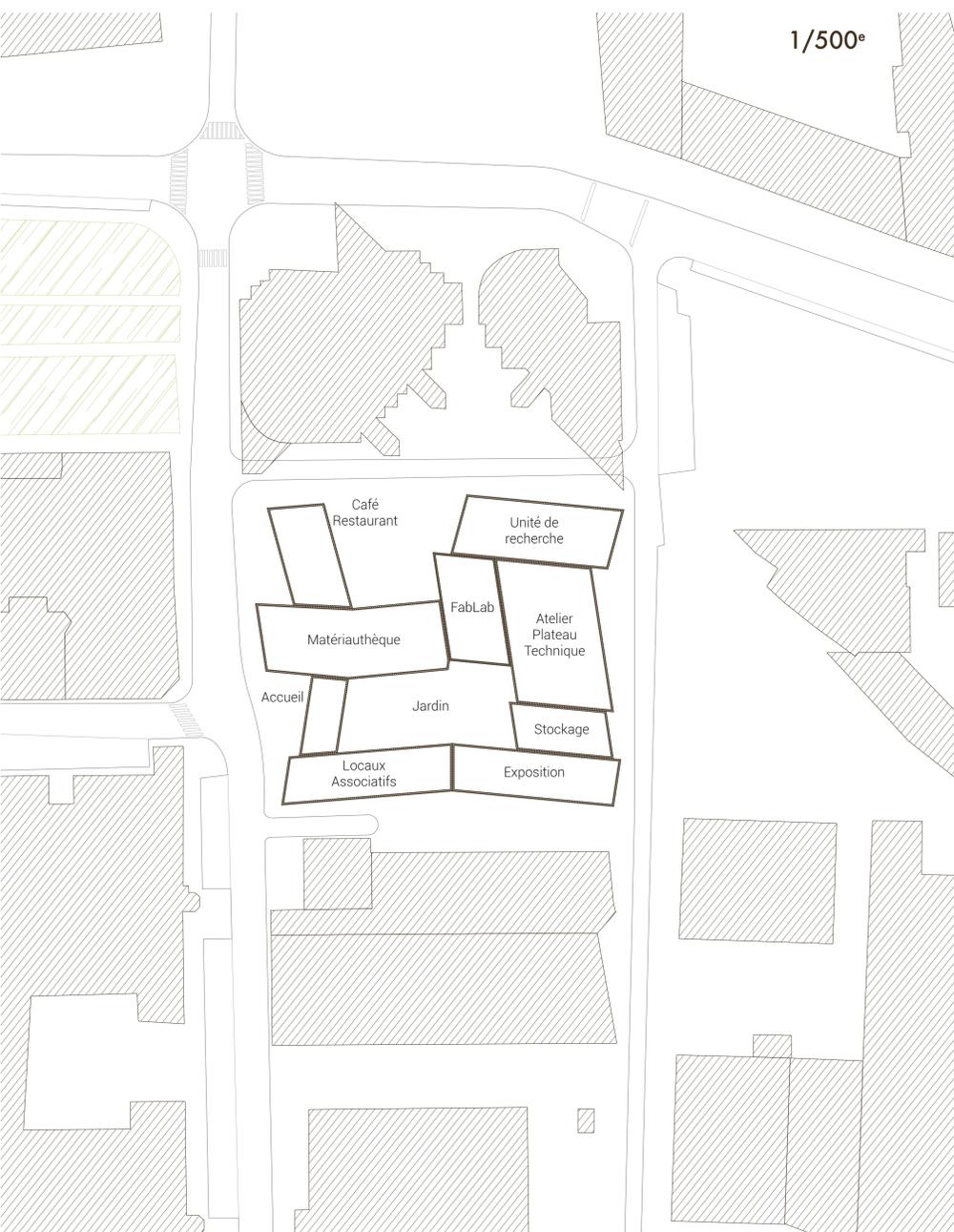


La Plaine Saint-Denis - Vue Générale



Site N°1 : Abandonnée. Finalement incompatible avec la philosophie initiale.

Site N°2. En cohérence cette fois.



Situation - Le site d'implantation Encyclopédie / Métallurgie



Esquisse