
Référentiel des bonnes pratiques d'écoconception des services numériques

Fiches des bonnes pratiques

Auteur: Pôle emploi



2023-08-07

Table des matières

1	Référentiel des bonnes pratiques d'écoconception des services numériques	1
2	Contributeurs	2
2.1	Auteurs	2
2.2	Contacts	2
2.3	Remerciments	2
3	Présentation	3
3.1	Public visé	3
3.2	Objectifs	3
4	Introduction	4
4.1	Le numérique responsable	4
5	Introduction	5
5.1	L'ÉCOCONCEPTION DES SERVICES NUMÉRIQUES	5
6	Introduction	6
6.1	LES ENJEUX DE L'ÉCOCONCEPTION	6
6.2	LES FORCES DE PÔLE EMPLOI	6
7	Introduction	8
7.1	LA DÉMARCHE ADOPTÉE PAR PÔLE EMPLOI	8
7.2	Formations	9
8	PRÉSENTATION DU RÉFÉRENTIEL	10
9	Présentation	11
9.1	FICHES PRATIQUES	11
10	Présentation	12
10.1	DESCRIPTION D'UNE FICHE PRATIQUE	12

11 Présentation	13
11.1 DESCRIPTION D'UNE FICHE PRATIQUE	13
12 Présentation	14
12.1 OUTILS PRÉCONISÉS AU 01/06/2023	14
13 Présentation	15
13.1 Sommaire	15
14 Equipe	16
15 Stratégie	17
16 UX	18
17 UI	19
18 Code	20
19 Architecture	21
19.1 Sous familles traitées dans ce chapitre :	21
20 Conclusion	22
20.1 contacts :	22
21 EQ-01 - Sensibiliser les parties prenantes à l'écoconception	23
21.1 Description :	23
21.2 Mise en œuvre :	23
21.3 Tests d'acceptance :	24
21.4 Mesures :	24
22 EQ-02 - Former les parties prenantes à l'écoconception	25
22.1 Description :	25
22.2 Mise en œuvre :	25
22.3 Tests d'acceptance :	25
23 EQ-03 - Désigner un référent écoconception	26
23.1 Description :	26
23.2 Mise en œuvre :	26
23.3 Tests d'acceptance :	26

24 ST-01 - Questionner la valeur du service numérique au vu de son impact environnemental	27
24.1 Description :	27
24.2 Mise en œuvre :	27
24.3 Tests d'acceptance :	28
25 ST-02 - Documenter le produit tout au long de son cycle de vie	29
25.1 Description :	29
25.2 Mise en œuvre :	29
25.3 Tests d'acceptance :	30
26 ST-03 - Définir une stratégie de décommissionnement	31
26.1 Description :	31
26.2 Mise en œuvre :	31
26.3 Tests d'acceptance :	32
27 ST-04 - Intégrer l'écoconception dans la backlog de l'équipe	33
27.1 Description :	33
27.2 Mise en œuvre :	33
27.3 Tests d'acceptance :	34
28 ST-05 - Organiser des temps dédiés à l'écoconception	35
28.1 Description :	35
28.2 Mise en œuvre :	35
28.3 Tests d'acceptance :	35
29 ST-06 - Systématiser des revues du service numérique	36
29.1 Description :	36
29.2 Mise en œuvre :	36
29.3 Tests d'acceptance :	36
30 ST-07 - Définir une stratégie d'archivage et de suppression des contenus obsolètes	37
30.1 Description :	37
30.2 Mise en œuvre :	37
30.3 Tests d'acceptance :	38
31 ST-08 - Imposer des exigences d'écoconception aux fournisseurs de services numériques	39
31.1 Description :	39
31.2 Mise en œuvre :	39
31.3 Tests d'acceptance :	39

32 ST-09 - Fixer des objectifs de réduction de l'empreinte environnementale du service	40
32.1 Description :	40
32.2 Mise en œuvre :	40
32.3 Tests d'acceptance :	40
33 ST-10 - Mesurer les impacts environnementaux du service tout au long de son cycle de vie	41
33.1 Description :	41
33.2 Mise en œuvre :	41
33.3 Tests d'acceptance :	41
34 ST-11 - Suivre les métriques fournies par les Core Web Vitals	42
34.1 Description :	42
34.2 Mise en œuvre :	43
34.3 Tests d'acceptance :	43
35 ST-12 - Mesurer l'expérience utilisateur	44
35.1 Description :	44
35.2 Mise en œuvre :	44
35.3 Tests d'acceptance :	45
36 ST-13 - Être attentif aux besoins des utilisateurs et vérifier l'apport de valeur du produit	46
36.1 Description :	46
36.2 Mise en œuvre :	46
36.3 Tests d'acceptance :	46
37 ST-14 - Maîtriser la taille du DOM	48
37.1 Description :	48
37.2 Mise en œuvre :	48
37.3 Tests d'acceptance :	49
38 ST-15 - Limiter le nombre de requêtes	50
38.1 Description :	50
38.2 Mise en œuvre :	50
38.3 Tests d'acceptance :	50
39 ST-16 - Établir et respecter un budget environnemental	51
39.1 Description :	51
39.2 Mise en œuvre :	51
39.3 Tests d'acceptance :	52

40 ST-17 - Vérifier la compatibilité du service numérique avec des faibles débits	53
40.1 Description :	53
40.2 Mise en œuvre :	53
40.3 Tests d'acceptance :	54
41 UX-01 - Faire en sorte que la page s'adapte à tous les terminaux d'affichage	55
41.1 Description :	55
41.2 Mise en œuvre :	55
41.3 Tests d'acceptance :	55
42 UX-02 - Adopter une approche mobile first si cela est pertinent	56
42.1 Description :	56
42.2 Mise en œuvre :	56
42.3 Tests d'acceptance :	57
43 UX-03 - Fournir un moyen de contrôle à l'utilisateur sur les contenus et services	58
43.1 Description :	58
43.2 Mise en œuvre :	58
43.3 Tests d'acceptance :	59
44 UX-04 - Bien connaître les différents segments utilisateurs et leurs problèmes	60
44.1 Description :	60
44.2 Mise en œuvre :	60
44.3 Tests d'acceptance :	61
45 UX-05 - Ne garder que les fonctionnalités essentielles	62
45.1 Description :	62
45.2 Mise en œuvre :	62
45.3 Tests d'acceptance :	63
46 UX-06 - Implémenter les fonctionnalités de manière sobre	64
46.1 Description :	64
46.2 Mise en œuvre :	64
46.3 Tests d'acceptance :	64
47 UX-07 - Indiquer les impacts environnementaux des fonctionnalités très consommatrices	66
47.1 Description :	66
47.2 Mise en œuvre :	66
47.3 Tests d'acceptance :	67

48 UX-08 - Concevoir un parcours utilisateur simple et efficace	68
48.1 Description :	68
48.2 Mise en œuvre :	68
48.3 Tests d'acceptance :	69
49 UX-09 - Favoriser un parcours utilisateur au sein d'un seul onglet	70
49.1 Description :	70
49.2 Mise en œuvre :	70
49.3 Tests d'acceptance :	71
50 UX-10 - Limiter l'ouverture de pop-ups	72
50.1 Description :	72
50.2 Mise en œuvre :	72
50.3 Tests d'acceptance :	72
51 UX-11 - Implémenter un moteur de recherche interne	73
51.1 Description :	73
51.2 Mise en œuvre :	73
51.3 Tests d'acceptance :	73
52 UX-12 - Permettre à l'utilisateur de relancer une recherche depuis sa page de résultats	74
52.1 Description :	74
52.2 Mise en œuvre :	74
52.3 Tests d'acceptance :	74
53 UX-13 - Indiquer au début d'un parcours utilisateur les données et documents qui seront demandés	75
53.1 Description :	75
53.2 Mise en œuvre :	75
53.3 Tests d'acceptance :	76
54 UX-14 - Indiquer à l'utilisateur lorsqu'un traitement en arrière plan est en cours	77
54.1 Description :	77
54.2 Mise en œuvre :	77
54.3 Tests d'acceptance :	78
55 UX-15 - Minimiser les notifications	79
55.1 Description :	79
55.2 Mise en œuvre :	79
55.3 Tests d'acceptance :	80

56 UX-16 - Permettre à l'utilisateur de régler les notifications	81
56.1 Description :	81
56.2 Mise en œuvre :	81
56.3 Tests d'acceptance :	81
56.4 Mesures :	82
57 UX-17 - Minimiser le poids des e-mails	83
57.1 Description :	83
57.2 Mise en œuvre :	83
57.3 Tests d'acceptance :	83
58 UX-18 - Choisir un service tiers écoconçu	84
58.1 Description :	84
58.2 Mise en œuvre :	84
58.3 Tests d'acceptance :	84
59 UX-19 - Désactiver les services tiers non nécessaires sauf demande explicite de l'utilisateur	85
59.1 Description :	85
59.2 Mise en œuvre :	85
59.3 Tests d'acceptance :	86
60 UX-20 - Permettre à l'utilisateur de refuser facilement les mécanismes de consentement	87
60.1 Description :	87
60.2 Mise en œuvre :	87
60.3 Tests d'acceptance :	88
61 UX-21 - Maintenir le sitemap à jour	89
61.1 Description :	89
61.2 Mise en œuvre :	89
61.3 Tests d'acceptance :	90
62 UI-01 - Utiliser un mécanisme de lazy loading	91
62.1 Description :	91
62.2 Mise en œuvre :	91
62.3 Tests d'acceptance :	91
63 UI-02 - Éviter le défilement de contenus	92
63.1 Description :	92
63.2 Mise en œuvre :	92
63.3 Tests d'acceptance :	92

64 UI-03 - Fournir des CSS print optimisés	93
64.1 Description :	93
64.2 Mise en œuvre :	93
64.3 Tests d'acceptance :	94
65 UI-04 - Minimiser le nombre de vidéos par page	95
65.1 Description :	95
65.2 Mise en œuvre :	95
65.3 Tests d'acceptance :	95
66 UI-05 - Raccourcir la durée des vidéos	96
66.1 Description :	96
66.2 Mise en œuvre :	96
66.3 Tests d'acceptance :	96
67 UI-06 - Optimiser les vidéos	98
67.1 Description :	98
67.2 Mise en œuvre :	99
67.3 Tests d'acceptance :	99
68 UI-07 - Arrêter la lecture des vidéos hors du cadre de visualisation	100
68.1 Description :	100
68.2 Mise en œuvre :	100
68.3 Tests d'acceptance :	101
69 UI-08 - Désactiver l'autoplay des vidéos	102
69.1 Description :	102
69.2 Mise en œuvre :	102
69.3 Tests d'acceptance :	103
70 UI-09 - Ne pas jouer les vidéos en boucle	104
70.1 Description :	104
70.2 Mise en œuvre :	104
70.3 Tests d'acceptance :	105
71 UI-10 - Prévoir une alternative texte aux vidéos	106
71.1 Description :	106
71.2 Mise en œuvre :	106
71.3 Tests d'acceptance :	107

72 UI-11 - Afficher la durée des contenus vidéo	108
72.1 Description :	108
72.2 Mise en œuvre :	108
72.3 Tests d'acceptance :	108
73 UI-12 - Minimiser le nombre d'audios par page	109
73.1 Description :	109
73.2 Mise en œuvre :	109
73.3 Tests d'acceptance :	109
74 UI-13 - Désactiver l'autoplay des audios	110
74.1 Description :	110
74.2 Mise en œuvre :	110
74.3 Tests d'acceptance :	110
75 UI-14 - Utiliser des formats d'audios optimisés	111
75.1 Description :	111
75.2 Mise en œuvre :	111
75.3 Tests d'acceptance :	111
76 UI-15 - Ajuster le niveau de compression des audios	112
76.1 Description :	112
76.2 Mise en œuvre :	112
76.3 Tests d'acceptance :	112
77 UI-16 - Ajuster le niveau de compression des audios	113
77.1 Description :	113
77.2 Mise en œuvre :	113
77.3 Tests d'acceptance :	114
78 UI-17 - Minimiser le nombre d'images	115
78.1 Description :	115
78.2 Mise en œuvre :	115
78.3 Tests d'acceptance :	116
79 UI-18 - Préférer le CSS aux images pour les éléments d'interface	117
79.1 Description :	117
79.2 Mise en œuvre :	117
79.3 Tests d'acceptance :	117

80 UI-19 - Générer des spritesheets CSS lorsque le contexte le permet	118
80.1 Description :	118
80.2 Mise en œuvre :	118
80.3 Tests d'acceptance :	118
81 UI-20 - Minimiser les dimensions des images	119
81.1 Description :	119
81.2 Mise en œuvre :	119
81.3 Tests d'acceptance :	119
81.4 Mesures :	120
82 UI-21 - Compresser et ajuster la résolution des images	121
82.1 Description :	121
82.2 Mise en œuvre :	121
82.3 Tests d'acceptance :	122
83 UI-22 - Fournir des formats d'images optimisés	123
83.1 Description :	123
83.2 Mise en œuvre :	123
83.3 Tests d'acceptance :	124
83.4 Mesures :	124
84 UI-23 - Proposer plusieurs tailles d'images	125
84.1 Description :	125
84.2 Mise en œuvre :	125
84.3 Tests d'acceptance :	126
85 UI-24 - Limiter les animations Javascript/CSS	127
85.1 Description :	127
85.2 Mise en œuvre :	127
85.3 Tests d'acceptance :	128
86 UI-25 - Limiter les carrousels automatiques	129
86.1 Description :	129
86.2 Mise en œuvre :	129
86.3 Tests d'acceptance :	129
87 UI-26 - Limiter le nombre de webfonts utilisées	131
87.1 Description :	131
87.2 Mise en œuvre :	131

87.3 Tests d'acceptance :	132
88 UI-27 - Utiliser les formats de police WOFF2 et WOFF	133
88.1 Description :	133
88.2 Mise en œuvre :	133
88.3 Tests d'acceptance :	134
89 UI-28 - Optimiser les documents	135
89.1 Description :	135
89.2 Mise en œuvre :	135
89.3 Tests d'acceptance :	136
90 UI-29 - Remplacer la complétion automatique par une alternative induisant moins de requêtes	137
90.1 Description :	137
90.2 Mise en œuvre :	137
90.3 Tests d'acceptance :	138
91 UI-30 - Limiter le poids et le format des fichiers que l'utilisateur doit téléverser	139
91.1 Description :	139
91.2 Mise en œuvre :	139
91.3 Tests d'acceptance :	140
92 UI-31 - Vérifier les formats attendus de la saisie de l'utilisateur avant soumission du formulaire	141
92.1 Description :	141
92.2 Mise en œuvre :	141
92.3 Tests d'acceptance :	142
93 CO-01 - Concaténer les fichiers CSS	143
93.1 Description :	143
93.2 Mise en œuvre :	143
93.3 Tests d'acceptance :	143
94 CO-02 - Utiliser uniquement les portions indispensables des frameworks CSS	144
94.1 Description :	144
94.2 Mise en œuvre :	144
94.3 Tests d'acceptance :	144

95 CO-03 - Utiliser des sélecteurs CSS efficaces	145
95.1 Description :	145
95.2 Mise en œuvre :	145
95.3 Tests d'acceptance :	146
96 CO-04 - Grouper les déclarations CSS similaires	147
96.1 Description :	147
96.2 Mise en œuvre :	147
96.3 Tests d'acceptance :	148
97 CO-05 - Modifier plusieurs propriétés CSS en une seule fois	149
97.1 Description :	149
97.2 Mise en œuvre :	149
97.3 Tests d'acceptance :	150
98 CO-06 - Prévenir le navigateur du déclenchement d'une animation	151
98.1 Description :	151
98.2 Mise en œuvre :	152
98.3 Tests d'acceptance :	152
99 CO-07 - Concaténer les fichiers Javascript	153
99.1 Description :	153
99.2 Mise en œuvre :	153
99.3 Tests d'acceptance :	153
100CO-08 - Utiliser uniquement les portions indispensables des librairies JavaScript	154
100.1Description :	154
100.2Mise en œuvre :	154
100.3Tests d'acceptance :	155
101CO-09 - Utiliser un framework javascript adapté et efficient	156
101.1Description :	156
101.2Mise en œuvre :	156
101.3Tests d'acceptance :	157
102CO-10 - Ne pas copier des tableaux via des boucles for, while, do..while et foreach	158
102.1Description :	158
102.2Mise en œuvre :	158
102.3Tests d'acceptance :	158

103CO-11 - Éviter d'utiliser SELECT * FROM	159
103.1 Description :	159
103.2 Mise en œuvre :	159
103.3 Tests d'acceptance :	160
104CO-12 - Éviter d'utiliser « EXECUTE IMMEDIATE »	161
104.1 Description :	161
104.2 Mise en œuvre :	161
104.3 Tests d'acceptance :	161
105CO-13 - Eviter de faire des requêtes SQL à l'intérieur d'une boucle	162
105.1 Description :	162
105.2 Mise en œuvre :	162
105.3 Tests d'acceptance :	163
106CO-14 - Ne pas réécrire les getter et setter natifs	164
106.1 Description :	164
106.2 Mise en œuvre :	164
106.3 Tests d'acceptance :	165
107CO-15 - Privilégier les variables locales	166
107.1 Description :	166
107.2 Mise en œuvre :	166
107.3 Tests d'acceptance :	167
108CO-16 - Supprimer le code inutilisé	168
108.1 Description :	168
108.2 Mise en œuvre :	168
108.3 Tests d'acceptance :	169
109CO-17 - Minifier le code	170
109.1 Description :	170
109.2 Mise en œuvre :	170
109.3 Tests d'acceptance :	170
110CO-18 - Ecrire un code propre	171
110.1 Description :	171
110.2 Mise en œuvre :	171
110.3 Tests d'acceptance :	172

111CO-19- Utiliser le validateur du W3C	173
111.1 Description :	173
111.2 Mise en œuvre :	173
111.3 Tests d'acceptance :	173
112CO-20- Valider le code avec un linter	174
112.1 Description :	174
112.2 Mise en œuvre :	174
112.3 Tests d'acceptance :	174
113CO-21- Utiliser des outils d'analyse de code statique	175
113.1 Description :	175
113.2 Mise en œuvre :	175
113.3 Tests d'acceptance :	175
114CO-22- Utiliser des outils d'analyse de code dynamique	176
114.1 Description :	176
114.2 Mise en œuvre :	176
114.3 Tests d'acceptance :	177
115CO-23- Vérifier les erreurs de code dans la console	178
115.1 Description :	178
115.2 Mise en œuvre :	178
115.3 Tests d'acceptance :	178
116CO-24- Réutiliser le code développé en interne	179
116.1 Description :	179
116.2 Mise en œuvre :	180
116.3 Tests d'acceptance :	180
117CO-25- Utiliser l'open source de manière responsable	181
117.1 Description :	181
117.2 Mise en œuvre :	182
117.3 Tests d'acceptance :	182
118CO-26- Utiliser la CI/CD de manière responsable	183
118.1 Description :	183
118.2 Mise en œuvre :	183
118.3 Tests d'acceptance :	184

119AR-01- Utiliser la CI/CD de manière responsable	185
119.1 Description :	185
119.2 Mise en œuvre :	186
119.3 Tests d'acceptance :	186
120AR-02- Vérifier l'existence des données avant de créer une structure de données ad hoc	187
120.1 Description :	187
120.2 Mise en œuvre :	187
120.3 Tests d'acceptance :	188
121AR-03- Réutiliser les datasets	189
121.1 Description :	189
121.2 Mise en œuvre :	190
121.3 Tests d'acceptance :	190
122AR-04- Définir une politique de cycle de vie des données et l'appliquer	191
122.1 Description :	191
122.2 Mise en œuvre :	192
122.3 Tests d'acceptance :	192
123AR-05- Optimiser la structure du modèle de données	193
123.1 Description :	193
123.2 Mise en œuvre :	193
123.3 Tests d'acceptance :	194
124AR-06- Optimiser les requêtes et les accès aux données au strict nécessaire	195
124.1 Description :	195
124.2 Mise en œuvre :	195
124.3 Tests d'acceptance :	196
125AR-07- Utiliser un format de données adapté (1/2)	197
125.1 Description :	197
126AR-07- Utiliser un format de données adapté (2/2)	200
126.1 Mise en œuvre :	200
126.2 Tests d'acceptance :	200
127AR-08- Utiliser un système de stockage adapté au besoin	201
127.1 Description :	201
127.2 Mise en œuvre :	201

127.3 Tests d'acceptance :	202
128AR-09- Utiliser des mécanismes de déduplication des données	203
128.1 Description :	203
128.2 Mise en œuvre :	203
128.3 Tests d'acceptance :	204
129AR-10- Utiliser un système de stockage adapté au besoin	205
129.1 Description :	205
129.2 Mise en œuvre :	205
129.3 Tests d'acceptance :	206
130AR-11- Centraliser les données de log, de trace et de monitoring	207
130.1 Description :	207
130.2 Mise en œuvre :	208
130.3 Tests d'acceptance :	208
131AR-12 - Alimenter les datacenters avec de l'électricité d'origine renouvelable ou bas carbone	209
131.1 Description :	209
131.2 Mise en œuvre :	209
131.3 Tests d'acceptance :	210
132AR-13 - Réutiliser la chaleur fatale des Datacenters	211
132.1 Description :	211
132.2 Mise en œuvre :	211
133AR-14 - Organiser les baies en allées chaudes et froides dans les centres de données	212
133.1 Description :	212
133.2 Mise en œuvre :	213
134AR-15 - Redimensionner les VM lorsque nécessaire	214
134.1 Description :	214
134.2 Mise en œuvre :	214
134.3 Tests d'acceptance :	215
135AR-16 - Maximiser l'utilisation des serveurs	216
135.1 Description :	216
135.2 Mise en œuvre :	216

136AR-17 - Privilégier une architecture Cloud native	217
136.1 Description :	217
136.2 Mise en œuvre :	217
136.3 Mise en œuvre :	218
137AR-18 - Opter pour une architecture capable de passer à l'échelle automatiquement	219
137.1 Description :	219
137.2 Mise en œuvre :	220
138AR-19 - Utiliser les bons outils hardware	221
138.1 Description :	221
138.2 Mise en œuvre :	222
138.3 Tests d'acceptance	222
139AR-19 - Utiliser les bons outils hardware	223
139.1 Description :	223
139.2 Mise en œuvre :	224
139.3 Tests d'acceptance	224
140AR-21 - Maximiser l'utilisation des caches HTTP	225
140.1 Description :	225
140.2 Mise en œuvre :	225
140.3 Tests d'acceptance	226
141AR-22 - Compresser les flux HTTP	227
141.1 Description :	227
141.2 Mise en œuvre :	227
141.3 Tests d'acceptance	228
142AR-23 - Protocole de transport (1/2)	229
142.1 Description :	229
142.2 Mise en œuvre :	229
142.3 Tests d'acceptance	230
143AR-23 - Protocole de transport (2/2)	231
143.1 Description :	231
143.2 Mise en œuvre :	231
143.3 Tests d'acceptance	231

144AR-24 - Utiliser un protocole d'échange adapté au contenu transféré	232
144.1 Description :	232
144.2 Mise en œuvre :	233
145AR-25 - Limiter les données stockées	234
145.1 Description :	234
145.2 Mise en œuvre :	234
145.3 Tests d'acceptance	235
146AR-26 - Privilégier les technologies standard interopérables	236
146.1 Description :	236
146.2 Mise en œuvre :	237
146.3 Tests d'acceptance	237

Table des figures

Liste des tableaux

1 Référentiel des bonnes pratiques d'écoconception des services numériques

EN COURS DE CONSTRUCTION ## MISE À JOUR 01/06/2023

2 Contributeurs

2.1 Auteurs

- Vincent RABAH, porteur de l'orientation Numérique responsable (DSI - ASIT)
- Sandra CHARRIER, responsable du département RSE (DSI - ASIT)
- Florian CHAMPIE, développeur (DSI - ASIT)
- Alan GUITARD, prestataire Sopra Steria
- Jessica VOLCKMANN, prestataire EY
- Clémence MODESTE, prestataire EY

2.2 Contacts

- Vincent RABAH - vincent.rabah@pole-emploi.fr
- Sandra CHARRIER - sandra.charrier@pole-emploi.fr

2.3 Remerciements

Les auteurs tiennent à remercier les personnes ayant participé à la construction du référentiel par leurs conseils et commentaires : Sébastien ASTORI, Sébastien GOYER, Violaine BOISSEAU, Vincent LEMEUNIER, Shirley NOEL, Thierry LEJEALLE, Patrick MAHU, Sébastien FRERE, Sophie BOURDA, Sylvain MAUGIRON, Gary GRANDJEAN, Sylvain POIRIER, Sabrina GOMEZ, Florian BOBO, Anais HOAREAU, Michel BEZY, Jean Bernard MARQUAIS, Samuel BORELL, Laurent GUINARD, Lionel GOUZY et Vincent LEGAULT

3 Présentation

3.1 Public visé

Ce référentiel s'adresse à l'ensemble des parties prenantes intervenant dans le cycle de vie d'un service numérique, par exemple :

- Le métier, qui exprime ses besoins
- Le Product Owner, qui traduit les besoins en exigences fonctionnelles
- Les UX/UI designers, qui conçoivent l'interface utilisateur
- Les intégrateurs et développeurs, qui traduisent les maquettes en code
- Les Ops, qui s'assurent de la disponibilité du service numérique
- Et toute personne intervenant dans le cycle de vie du service numérique

3.2 Objectifs

Sensibiliser les parties prenantes à l'écoconception des services numériques.

Fournir un ensemble de bonnes pratiques à appliquer afin de limiter l'impact environnemental d'un service numérique tout au long de son cycle de vie.

4 Introduction

4.1 Le numérique responsable

L'apparente immatérialité du Numérique ne permet pas d'envisager directement ses impacts environnementaux. Pourtant, l'infrastructure sur laquelle se basent nos usages numériques est bien réelle : au-delà des terminaux (ordinateurs, téléphones, etc...), elle est aussi composée d'équipements de réseaux (câbles, routeurs, répéteurs, etc...) et de centres de données.

En décembre 2021, le Système d'Informations de Pôle emploi représentait : les Chiffres clés

Aujourd'hui, la part du Numérique dans les émissions de gaz à effet de serre mondiales est estimée à 4 %. Les impacts environnementaux du Numérique sont nombreux : dérèglement climatique, consommation en eau et en énergie, épuisement de ressources non renouvelables, pollution, perte de la biodiversité, etc. Ces impacts risquent de s'aggraver si les individus et les collectivités n'agissent pas.

Pour cette raison, Pôle emploi s'est engagé dans une démarche d'amélioration continue visant à réduire l'empreinte écologique, économique et sociale des technologies de l'information et de la communication en signant une charte Numérique Responsable en collaboration avec l'Institut du Numérique Responsable (INR). Depuis, plusieurs projets Numérique Responsable ont été mis en place comme par exemple : le déploiement d'un kit de sensibilisation au Numérique Responsable, la mise en place d'une démarche de récupération de chaleur dans les datacenters, la mesure de la consommation énergétique des équipements numériques, la construction d'un référentiel d'écoconception des services numériques, etc.

Afin de concrétiser cet engagement, la DSI a placé le Numérique Responsable au cœur du programme de transformation de son SI, TESI 22. C'est dans le contexte que la démarche d'écoconception des services numériques de Pôle emploi a été lancée et que le présent référentiel a été réalisé.

5 Introduction

5.1 L'ÉCOCONCEPTION DES SERVICES NUMÉRIQUES

L'écoconception est une démarche d'amélioration continue... - Multi étapes : elle prend en compte l'ensemble du cycle de vie du service numérique - Multi composants : elle adresse l'ensemble des couches techniques d'un service numérique - Multi acteurs : elle mobilise l'ensemble des parties prenantes du service numérique - Multi critères : elle mesure l'ensemble des impacts environnementaux afin de limiter les transferts d'impacts

6 Introduction

6.1 LES ENJEUX DE L'ÉCOCONCEPTION

Les avantages de l'écoconception des services numériques ne se limitent pas qu'aux bénéfices environnementaux :

- Optimisation des coûts : en concevant des services moins consommateurs en ressources, l'écoconception permet de réaliser des économies pour l'entreprise en rationalisant le parc matériel, ou encore en diminuant les frais d'hébergement.
- Gain de performance : Un service numérique écoconçu nécessite peu de ressources pour fonctionner, il est donc rapide et performant.
- Amélioration de l'utilisabilité : Une interface graphique simple et épurée favorise sa compréhension et sa manipulation.

L'écoconception apporte aussi d'autres avantages comme par exemple l'amélioration de l'image de marque, une différenciation marketing, une meilleure expérience utilisateur, un meilleur référencement web, l'augmentation de l'audience, etc.

6.2 LES FORCES DE PÔLE EMPLOI

Le contexte actuel de Pôle emploi est propice à la mise en place d'une démarche d'écoconception des services numériques :

- La DSI dispose d'une usine logicielle, lui permettant d'avoir une vue complète sur l'ensemble du cycle de vie d'un service numérique, de ses impacts et des parties prenantes impliquées.
- La DSI dispose d'un pôle RSE (au sein d'ASIT) et de nombreux référents (toute direction) qui soutiennent le projet
- En tant qu'administration du service public et Opérateur de Services Essentiels de l'état OSE, Pôle emploi se doit de donner l'exemple et a donc tout intérêt à écoconcevoir ses services numériques
- Pôle emploi s'est fortement engagé dans le Numérique Responsable

- L'orientation Numérique Responsable est au cœur de la transformation du SI

7 Introduction

7.1 LA DÉMARCHE ADOPTÉE PAR PÔLE EMPLOI

Un référentiel d'écoconception de Pôle emploi explicite l'ensemble de bonnes pratiques d'écoconception propres au contexte de Pôle emploi. Il a été construit sur la base des bonnes pratiques d'écoconception figurant dans les principaux référentiels d'écoconception du marché, comme par exemple :

- Le référentiel des 115 bonnes pratiques¹ du Collectif Green IT
- Le référentiel général d'écoconception de services numériques² de la DINUM, le MTE, l'ADEME et l'INR
- Le guide d'écoconception³ de services numériques des Designers éthiques

Plusieurs ateliers ont été organisés avec les parties prenantes de la DSI afin d'adapter les bonnes pratiques sélectionnées au contexte de Pôle emploi et rédiger les fiches pratiques explicitant les bonnes pratiques.

ORGANISATION DU RÉFÉRENTIEL

Le présent référentiel est organisé en 6 chapitres correspondant chacun aux composants d'un service numérique :

N°	Nom	Détails
1	Equipe	Connaissances de l'équipe vis-à-vis de l'écoconception
2	Stratégie	Suivi de la pertinence et des enjeux du service numérique tout au long de son cycle de vie
3	UX	Etude du besoin et élaboration des parcours utilisateur
4	UI	Elaboration de l'interface utilisateur

1. <https://github.com/cnumr/best-practices>

2. <https://ecoresponsable.numerique.gouv.fr/publications/referentiel-general-ecoconception/>

3. <https://eco-conception.designersethiques.org/guide/fr/>

N°	Nom	Détails
5	Code	Développement frontend et backend
6	Architecture	Articulation entre les couches techniques du service numérique

7.2 Formations

Des formations par métiers sont disponibles pour vous permettre d'appliquer la démarche et de vous guider dans l'utilisation du référentiel :

- Formation de Sensibilisation à l'écoconception - code : 95ECSE
- L'écoconception pour les Développeurs - code : 95ECDT

8 PRÉSENTATION DU RÉFÉRENTIEL

9 Présentation

9.1 FICHES PRATIQUES

Chaque bonne pratique est présentée sous la forme d'une fiche synthétique dont la structure est toujours identique.

10 Présentation

10.1 DESCRIPTION D'UNE FICHE PRATIQUE

Identifiant : code unique associé à la bonne pratique. Cet identifiant est indispensable pour :

- Assurer le suivi dans le temps de la bonne pratique ;
- Permettre de créer des sous-ensembles de ce référentiel ;
- Numéroté les bonnes pratiques en s'assurant de se référer toujours à la même bonne pratique

Titre : intitulé concis de la bonne pratique

Priorité : niveau de priorité de la bonne pratique :

Score	Priorité	Signification
3	Must have	La bonne pratique doit être mise en œuvre en priorité car elle a un potentiel de réduction de l'impact environnemental important et qu'elle est facile à mettre en œuvre, ou qu'elle est indispensable à la mise en œuvre d'autres pratiques clés.
2	Nice to have	La bonne pratique a un potentiel de réduction de l'impact environnemental moyen et pourrait être mise en œuvre sans difficulté majeure. Elle offre un bon compromis entre le temps passé et le résultat obtenu.
1	Should have	Bien que cette bonne pratique permette de limiter les besoins en ressources, le potentiel d'économie est faible ou la difficulté de mise en œuvre est importante.

Bénéfices : gains apportés par la mise en œuvre de la bonne pratique : | Score | Signification |
|:---:|:-----:| | Réduction de l'empreinte environnementale du service numérique | | Amélioration des performances du service numérique | | Amélioration de l'utilisabilité du service numérique | | Limitation de la contribution du service numérique à l'obsolescence |
| | Observabilité |

11 Présentation

11.1 DESCRIPTION D'UNE FICHE PRATIQUE

État : état de la bonne pratique vis-à-vis de Pôle emploi (à faire / à déployer / à poursuivre)

Responsable : personne qui en charge d'appliquer la bonne pratique (Product Owner, développeur, UX/UI designer, intégrateur, etc.)

Phase : phase du cycle de vie du service numérique durant laquelle la bonne pratique doit être mise en œuvre (build / run / cycle de vie entier)

Catégorie : les bonnes pratiques sont classées en familles, elles-mêmes divisées en sous familles :

Famille	Sous familles
Equipe	Connaissances
Stratégie	Démarche, pilotage et spécifications
UX	E-mail, fonctionnalités, notifications, parcours utilisateur, personas et SEO
UI	Contenus et interface
Code	CSS, Javascript, Java, SQL, pratiques transverse et méthodes et outils
Architecture	Infrastructure, principes d'architecture, données et réseau

Description : explication de la bonne pratique et de son contexte

Phase : étape du cycle de vie du service numérique où est appliquée la bonne pratique (Build / Run / Fin de vie)

Mise en œuvre : description concrète de la manière dont doit être mise en œuvre la bonne pratique par son responsable.

Test : description du/des test(s) à réaliser pour vérifier la bonne mise en œuvre de la bonne pratique

Mesure : mesure des gains engendrés par la mise en place de la bonne pratique

12 Présentation

12.1 OUTILS PRÉCONISÉS AU 01/06/2023

Calcul de l’empreinte environnementale (ST-06) : GreenIT Analysis ¹

Mesure du nombre d’éléments du DOM (ST-10) : GreenIT Analysis ²

Mesure du nombre de requêtes (ST-11) : Console ou GreenIT Analysis ³

Mesure du poids de la page (ST-12) : Console ou GreenIT Analysis ⁴

Tableau comparatif des manières d’implémenter une fonctionnalité (UX-03) : En cours de construction

1. <https://chrome.google.com/webstore/detail/greenit-analysis/mofbfhffeklkbebfclfaiifefjflcpad?hl=fr>

2. <https://chrome.google.com/webstore/detail/greenit-analysis/mofbfhffeklkbebfclfaiifefjflcpad?hl=fr>

3. <https://chrome.google.com/webstore/detail/greenit-analysis/mofbfhffeklkbebfclfaiifefjflcpad?hl=fr>

4. <https://chrome.google.com/webstore/detail/greenit-analysis/mofbfhffeklkbebfclfaiifefjflcpad?hl=fr>

13 Présentation

13.1 Sommaire

1. Equipe
 - Connaissances
 - Rôles
2. Stratégie
 - Démarche
 - Pilotage
3. UX
 - Personas
 - Fonctionnalités
 - Parcours utilisateur
 - Messages d'informations
 - SEO
4. UI
 - Interface
 - Contenus
5. Code
 - CSS
 - Javascript
 - Java
 - Transverse
 - Méthodes et outils
6. Architecture
 - Infrastructure
 - Données
 - Réseau

14 Equipe

Un service numérique, même s'il repose sur des technologies et une infrastructure matérielle, est avant tout le fruit du travail mené par une équipe. Si cette équipe n'a pas de connaissances sur la notion d'écoconception, il est fort probable que le service numérique soit construit sans prendre en considération les impacts environnementaux qui pourront être induits lors de son utilisation. Ainsi, l'équipe doit nécessairement être sensibilisée et formée à l'écoconception pour en connaître les grands principes et bonnes pratiques, et pouvoir les appliquer.

Bonnes pratiques traitées dans ce chapitre :

Sensibiliser les parties prenantes à l'écoconception

Former les parties prenantes à l'écoconception

Désigner un référent écoconception

Des formations par métiers sont disponibles pour vous permettre d'appliquer la démarche et de vous guider dans l'utilisation du référentiel :

- Formation de Sensibilisation à l'écoconception - code : 95ECSE

15 Stratégie

Axer la stratégie mise en place autour d'un service numérique sur la réduction de ses impacts environnementaux tout au long de son cycle de vie est un premier pas vers l'écoconception. Les leviers que nous avons identifiés sont notamment :

Mettre en place des méthodes de travail qui mettent les utilisateurs au centre du projet, favorisent la communication et permettent plus globalement d'améliorer en continu le service numérique.

Piloter le cycle de vie du service numérique, en particulier suivre sa consommation de ressources, l'utilisation de ses fonctionnalités, ses composants, etc. et décider au besoin de son décommissionnement.

Des formations par métiers sont disponibles pour vous permettre d'appliquer la démarche et de vous guider dans l'utilisation du référentiel :

- Formation de Sensibilisation à l'écoconception - code : 95ECSE
- L'écoconception pour les Développeurs - code : 95ECDT
- Formation UX/UI - code : XXXX

16 UX

Les choix effectués vis-à-vis du parcours utilisateur (appelé aussi « User Experience ») lors de la phase de conception du service numérique impactent directement l'empreinte environnementale qui lui est associée. En effet, d'une manière générale, l'élément qui pollue le moins est celui qui n'existe pas. Cette règle s'applique par extension aux fonctionnalités du service, d'autant plus que 70 % des fonctionnalités demandées par les utilisateurs ne sont jamais, ou rarement, utilisées. L'enjeu, ici, sera de mieux déterminer le besoin métier, afin de développer uniquement les fonctionnalités essentielles.

Par ailleurs, un des objectifs de l'écoconception est qu'un service numérique monopolise le moins longtemps possible le réseau et les serveurs afin de limiter la quantité de traitements et de données tout au long de la chaîne applicative et le temps passé par l'internaute devant son terminal. Un parcours utilisateur simple, fluide et efficace permettra de répondre à cet objectif.

Des formations par métiers sont disponibles pour vous permettre d'appliquer la démarche et de vous guider dans l'utilisation du référentiel :

- Formation de Sensibilisation à l'écoconception - code : 95ECSE
- Formation UX/UI - code : XXXX

17 UI

L'interface graphique (appelée aussi « User Interface ») est un levier important de réduction de l'empreinte environnementale d'un service numérique. Elle doit être sobre et épurée afin de faciliter sa compréhension et sa manipulation. Concrètement, la maquette ne doit présenter que des éléments qui apportent réellement de la valeur aux utilisateurs (des éléments qui correspondent aux règles des 3U : des éléments utiles, utilisables et utilisés). Ainsi, la maquette d'une interface utilisateur écoconçue doit limiter au maximum les éléments qui n'ont pas de valeur ajoutée pour l'utilisateur et ne pas retenir l'attention de l'utilisateur.

L'intégration, c'est-à-dire la traduction de la maquette en une interface graphique utilisateur fonctionnelle basée sur HTML, CSS et Javascript, donne lieu à une multitude de choix et paramètres qui auront des impacts plus ou moins importants sur la consommation de ressources du service numérique. Ainsi, les bonnes pratiques d'écoconception présentées dans ce chapitre donneront les clés aux intégrateurs et développeurs afin de faire un meilleur usage des contenus (vidéos, animations, audios, images, documents, formulaires, polices, etc).

Des formations par métiers sont disponibles pour vous permettre d'appliquer la démarche et de vous guider dans l'utilisation du référentiel :

- Formation de Sensibilisation à l'écoconception - code : 95ECSE
- Formation UX/UI - code : XXXX

18 Code

Dans ce chapitre, nous traiterons l'aspect du développement, en se focalisant sur les langages suivants : HTML, CSS, Javascript, Python et Java. Les pratiques principales seront avant tout :

D'écrire un code de qualité, afin de faciliter sa maintenance et ainsi limiter son obsolescence

D'optimiser le code, de sorte à ce qu'il mobilise moins de ressources informatiques pour fonctionner

Des formations par métiers sont disponibles pour vous permettre d'appliquer la démarche et de vous guider dans l'utilisation du référentiel :

- Formation de Sensibilisation à l'écoconception - code : 95ECSE
- L'écoconception pour les Développeurs - code : 95ECDT

19 Architecture

Un service numérique est composé de plusieurs couches techniques (design, code frontend, code backend, etc.). Dans ce chapitre, l'architecture se définit comme l'articulation des composants entre ces différentes couches techniques.

L'architecture se décline en grands principes, lignes directrices qui structurent l'organisation globale du service numérique. Elle est composée de données qui transitent entre les composants applicatifs via le réseau. Le tout repose sur une infrastructure matérielle (serveurs, unités de stockage, équipements de réseau et de télécommunication, etc.).

L'objectif des bonnes pratiques d'architecture est, avant tout, de :

- Rationnaliser et rallonger la durée de vie de l'infrastructure matérielle, pour limiter l'impact environnemental associé à la fabrication de nouveaux équipements.
- Minimiser le volume de données et optimiser leur échange, pour limiter les impacts liés à leur transport, transformation, stockage, etc.

19.1 Sous familles traitées dans ce chapitre :

- Infrastructure
- Principes d'architecture
- Données
- Réseau

20 Conclusion

Nous espérons que ce référentiel vous a été utile et a permis de mener à bien vos projets. N'hésitez pas à nous partager vos retours sur ce référentiel ici afin que nous puissions l'améliorer en continu.

20.1 contacts :

Vincent RABAH - vincent.rabah@pole-emploi.fr

Sandra CHARRIER - sandra.charrier@pole-emploi.fr

Readme à venir

21 EQ-01 - Sensibiliser les parties prenantes à l'écoconception

Catégorie : - Équipe - Connaissances

Phase : Build

Responsable : Équipe produit

Bénéfices : 1

Priorité : 3

21.1 Description :

Dans un contexte où la transformation digitale des entreprises est en plein essor, chaque geste compte. Aujourd'hui, 73% des français ne réalisent pas l'impact du digital sur l'environnement, il est donc essentiel de provoquer dès à présent une prise de conscience chez eux et leur donner les clés pour agir. Sachant qu'en 2021, les actifs français travaillaient en moyenne 36,9h au cours d'une semaine habituelle, les entreprises ont un rôle non négligeable à jouer dans la sensibilisation et l'éducation des collaborateurs sur les bons gestes à adopter dans leur quotidien.

21.2 Mise en œuvre :

En tant qu'équipe produit, je peux me sensibiliser à l'écoconception en : - Assistant aux sessions de sensibilisations proposées par ASIT - Consultant les différents supports mis à ma disposition (Référentiel d'architecture ¹, infographie, etc.)

1. <http://architecture.pole-emploi.intra/lecoconception/>

21.3 Tests d'acceptance :

100% des membres de l'équipe ont assisté à une session de sensibilisation sur le sujet de l'écoconception de services numériques.

21.4 Mesures :

A venir.

22 EQ-02 - Former les parties prenantes à l'écoconception

Catégorie : - Équipe - Connaissances

Phase : Build

Responsable : Équipe produit

Bénéfices : 1

Priorité : 3

22.1 Description :

Une fois les collaborateurs sensibilisés aux enjeux et bonnes pratiques clés de l'écoconception, il est nécessaire de leur fournir les moyens de mettre en œuvre ce qu'ils ont appris. Réaliser le plan de formation à l'écoconception est un moyen simple et efficace pour y parvenir. Par ailleurs, une documentation compréhensible, claire et détaillée sur la mise en œuvre concrète des bonnes pratiques d'écoconception doit être mise à disposition des collaborateurs. Également, un accompagnement de proximité peut aussi être mis en place pour guider les équipes qui le souhaitent et approfondir les bonnes pratiques.

22.2 Mise en œuvre :

En tant qu'équipe produit, je me forme à l'écoconception de services numériques en : - Assistant à une session de formation proposée par mon organisation - Prenant connaissance du référentiel d'écoconception de Pôle emploi

22.3 Tests d'acceptance :

100% des membres de l'équipe ont assisté à une session de formation à l'écoconception

23 EQ-03 - Désigner un référent écoconception

Catégorie : - Équipe - Rôles

Phase : Build

Responsable : Équipe produit

Bénéfices : 1

Priorité : 3

23.1 Description :

Il est essentiel qu'une personne soit désignée au sein de l'équipe pour porter le sujet de l'écoconception. Cela permet notamment de ne pas mobiliser toute l'équipe à chaque échange avec l'équipe écoconception (ASIT RSE), mais aussi de s'assurer que la dimension écoconception sera bel et bien prise en compte dans le projet. Le référent écoconception n'a pas besoin d'avoir de connaissances techniques sur l'écoconception. Tous les métiers peuvent porter cette responsabilité. D'usage, nous avons plutôt observé les profils associés à la conduite de projet (PO, PM, etc.) s'emparer de ce rôle, mais dans certains cas, des profils techniques (développeur) peuvent tout à fait le faire. L'important est que la personne référente soit sensible au sujet de l'écoconception. Cette bonne pratique s'est d'ailleurs révélée très importante lors des POCs d'écoconception menés fin 2022.

23.2 Mise en œuvre :

En tant qu'équipe produit, je me concerte pour désigner un référent écoconception.

23.3 Tests d'acceptance :

Un référent d'écoconception a été désigné par l'équipe

24 ST-01 - Questionner la valeur du service numérique au vu de son impact environnemental

Catégorie : - Stratégie - Démarche

Phase : Build

Responsable : Product Owner

Bénéfices : 1

Priorité : 3

24.1 Description :

Aujourd'hui, la création de projets numériques ou l'ajout de fonctionnalités est la conséquence de demande métier ou de la stratégie de l'entreprise. Un projet numérique est validé ou non selon des contraintes de budget, disponibilité des ressources, stratégie, etc. La dimension environnementale n'est pas prise en compte dans cette évaluation. Or, un service numérique induit intrinsèquement des impacts environnementaux. Dès lors, il est nécessaire de prendre en compte les impacts environnementaux (en plus des autres critères) de ce nouveau service numérique / cet ajout de fonctionnalités et de les confronter à la valeur réelle apportée. Cela permettra de statuer de la création ou non d'un projet.

24.2 Mise en œuvre :

En tant que Product Owner, j'utilise la matrice d'évaluation () afin de statuer de la création de mon projet numérique.

24.3 Tests d'acceptance :

- La matrice d'évaluation a été utilisée
- La création ou non du projet numérique a pris en compte le résultat obtenu avec la matrice

25 ST-02 - Documenter le produit tout au long de son cycle de vie

Catégorie : - Stratégie - Démarche

Phase : Cycle de vie entier

Responsable : Equipe produit

Bénéfices : 2

Priorité : 3

25.1 Description :

Afin de comprendre le service numérique, son écosystème et d'en maîtriser les impacts, une documentation claire, cohérente, précise et standardisée doit être rédigée par l'équipe produit. Ces éléments devront à la fois être transmis aux équipes de réalisation et archivés pour documenter le produit dans sa globalité. Cela facilitera la prise en main et l'évolution du service numérique par tout collaborateur non initié.

25.2 Mise en œuvre :

En tant qu'équipe produit, je renseigne, tout au long du cycle de vie du service numérique : - Le contexte dans lequel le service numérique a été créé (objectif, équipe, etc.) - Les dépendances du service numérique (services tiers utilisés, source des données, etc.) - Les choix techniques réalisés - L'ensemble des choix et arbitrages effectués vis-à-vis de l'écoconception ainsi que leurs justifications (ex : stratégie de décommissionnement)

25.3 Tests d'acceptance :

- L'écosystème du service numérique ainsi que les technologies utilisées sont renseignés dans un outil
- Les choix et arbitrages effectués vis-à-vis de l'écoconception et leurs justifications sont renseignés dans un outil

26 ST-03 - Définir une stratégie de décommissionnement

Catégorie : - Stratégie - Démarche

Phase : Build

Responsable : Equipe produit

Bénéfices : 2

Priorité : 3

26.1 Description :

Afin de ne pas laisser actifs des données, composants ou matériels utilisés par le service numérique non utilisé, il est important d'identifier et de prévoir les dépendances entre matériels et fonctionnalités déployés. L'identification des scénarios de fin d'usage ou de vie pour les composants matériels, logiciels et les données décommissionnés doit aussi être anticipée.

26.2 Mise en œuvre :

En tant qu'équipe produit, je définis une stratégie de fin de vie : - des données : décider de conserver les données en l'état, les détruire, les réaffecter à un autre usage, ou les archiver lors du décommissionnement du service sur politique de vie des données - des composants logiciels : pourra-t-on les réutiliser en interne ou les mettre à disposition en Open source ? - du matériel : le matériel pourrait être réemployé au sein de l'entreprise après le décommissionnement du service numérique Je rédige cette stratégie dans la documentation opérationnelle de la démarche.

26.3 Tests d'acceptance :

Une stratégie de décommissionnement a été élaborée, rédigée dans la documentation opérationnelle et partagée à toute l'équipe.

27 ST-04 - Intégrer l'écoconception dans la backlog de l'équipe

Catégorie : - Stratégie - Démarche

Phase : Cycle de vie entier

Responsable : Product Owner

Bénéfices : 1

Priorité : 3

27.1 Description :

Afin de garantir la prise en compte des enjeux d'écoconception par les équipes, il est conseillé d'ajouter les activités d'écoconception directement dans la backlog de l'équipe. Cela peut être réalisé de plusieurs façons : - Créer des features ou US spécifiques à l'écoconception - Ajouter une dimension écoconception à la formulation des US : « en tant que ... je veux ... afin de ... » + « en veillant à... » ou « en se limitant à ... » ou « en autorisant... » - Intégrer des critères d'écoconception dans la Definition of Done

Par ailleurs, un éco-score peut être utilisé pour prioriser les éléments de la backlog en plus de la valeur métier. Cet éco-score peut être calculé à partir de différents facteurs pondérés (méthode Klee Group) : - **Transfert** : volume de données circulant sur le réseau - **Stockage** : volume de données stockées sur serveur - **Maintenance** : facilité de maintenance de l'application - **Compatibilité** : variété des équipements permettant l'accès à l'application - **Parcours** : simplicité d'utilisation

27.2 Mise en œuvre :

En tant que Product Owner, je détermine la manière la plus adaptée d'intégrer l'écoconception à la backlog et construis la backlog en conséquence.

27.3 Tests d'acceptance :

La backlog contient des exigences d'écoconception.

28 ST-05 - Organiser des temps dédiés à l'écoconception

Catégorie : - Stratégie - Démarche

Phase : Cycle de vie entier

Responsable : Equipe produit

Bénéfices : 1

Priorité : 3

28.1 Description :

Afin de faciliter la prise en compte de l'écoconception par l'équipe, et de piloter la mise en œuvre de la démarche, il est essentiel d'organiser des créneaux dédiés à l'écoconception réunissant toute l'équipe.

28.2 Mise en œuvre :

En tant que Scrum Master, je m'assure que la thématique de l'écoconception soit abordée dans les cérémonies de l'équipe. En tant que Tech lead, j'inclus un temps relatif à l'écoconception dans les cérémonies que j'anime. En tant que Scrum Master, j'organise des points réguliers avec le référent écoconception pour lui présenter l'état d'avancement de la démarche et identifier d'autres leviers d'écoconception.

28.3 Tests d'acceptance :

- L'équipe aborde la thématique de l'écoconception au moins 1 fois par semaine lors d'une cérémonie
- Des points réguliers sont organisés avec le référent écoconception

29 ST-06 - Systématiser des revues du service numérique

Catégorie : - Stratégie - Démarche

Phase : Cycle de vie entier

Responsable : Equipe produit

Bénéfices : 1

Priorité : 3

29.1 Description :

Des revues sont des temps dédiés d'analyse du service numérique entre pairs. Elles sont l'occasion pour l'équipe de vérifier la qualité du service numérique. Dans le cadre de l'écoconception, elles permettent également de vérifier l'application des bonnes pratiques. Aussi, il conviendra d'organiser des revues régulières, par exemple à la fin de chacune des étapes du cycle de vie du service numérique (conception, développement, déploiement et fin de vie).

29.2 Mise en œuvre :

En tant qu'équipe produit, j'organise régulièrement des instances de revues de mon service numérique, au cours desquelles je vérifie les erreurs de code, j'anticipe les éventuels problèmes et les corrige. Je prends également un temps dédié pour vérifier que les choix de conception ont été respectés.

29.3 Tests d'acceptance :

Des revues du service numérique ont été organisées à chaque étape du cycle de vie du service numérique

30 ST-07 - Définir une stratégie d'archivage et de suppression des contenus obsolètes

Catégorie : - Stratégie - Démarche

Phase : Build

Responsable : Éditeur de contenus

Bénéfices : 1

Priorité : 3

30.1 Description :

91% de toutes les pages du web n'ont aucun trafic sur Google (Ahrefs, 2018). Dès lors, on peut s'interroger sur la valeur du contenu produit, et d'une manière plus générale, sur la stratégie éditoriale adoptée. Une stratégie éditoriale sobre consiste à : - Réduire la production de contenus à l'essentiel - Éviter le green washing - Adopter une diffusion raisonnée des contenus - Piloter le cycle de vie des contenus

30.2 Mise en œuvre :

En tant qu'éditeur de contenus : - Avant de concevoir et publier du contenu, j'évalue si cette publication va réellement apporter quelque chose à l'utilisateur / à mon entreprise. Je me demande également si cette publication ne relève pas du green washing (Quel est l'objectif de cette publication : servir l'utilisateur ou bien récolter l'adhésion des utilisateurs en communiquant sur des sujets environnementaux ?) - Je diffuse mon contenu sur les plateformes adaptées et à une fréquence raisonnée - Tout au long du cycle de vie du contenu, je surveille son utilisation et l'archive et/ou supprime s'il n'est pas ou peu consulté, ou obsolète. Cette stratégie peut être automatisée en définissant une date d'expiration et un processus d'archivage et/ou de purge automatique.

30.3 Tests d'acceptance :

Des revues du service numérique ont été organisées à chaque étape du cycle de vie du service numérique

31 ST-08 - Imposer des exigences d'écoconception aux fournisseurs de services numériques

Catégorie : - Stratégie - Démarche

Phase : Build

Responsable : Acheteur

Bénéfices : 1

Priorité : 3

31.1 Description :

Un projet est rarement réalisé en totalité avec des ressources internes : de nombreuses ressources externes sont bien souvent mobilisées. Dans ce contexte, elles se doivent d'être également alignées avec la démarche d'écoconception initiée par Pôle emploi. Actuellement, Pôle emploi (ASIT) travaille avec le département Achats et la DPIRS pour inclure des clauses environnementales dans les appels d'offres de services numériques.

31.2 Mise en œuvre :

En tant qu'acheteur, lorsque je lance un appel d'offres de service numérique, j'y inclue les critères d'écoconception établis par Pôle emploi et figurant dans le catalogue des clauses.

31.3 Tests d'acceptance :

Les critères d'écoconception établis par Pôle emploi ont été intégrés au processus d'appel d'offres.

32 ST-09 - Fixer des objectifs de réduction de l’empreinte environnementale du service

Catégorie : - Stratégie - Démarche

Phase : Build

Responsable : Equipe produit

Bénéfices : 1

Priorité : Optionnel

32.1 Description :

La démarche d'écoconception vise à réduire l'empreinte environnementale des services numériques sur lesquels elle est appliquée. Afin de définir une trajectoire d'optimisation, il est nécessaire de se fixer des objectifs de réduction de l'empreinte environnementale du service.

32.2 Mise en œuvre :

En tant qu'équipe produit, après avoir mesuré ma maturité d'écoconception, j'identifie les bonnes pratiques d'écoconception que je peux appliquer et estime (travail en cours par l'équipe écoconception), à partir de ce résultat, un objectif de réduction de l'empreinte environnementale du service ou de la page en question.

32.3 Tests d'acceptance :

Un objectif de réduction de l'empreinte environnementale du service ou de la page concernée a été défini.

33 ST-10 - Mesurer les impacts environnementaux du service tout au long de son cycle de vie

Catégorie : - Stratégie - Pilotage

Phase : Cycle de vie entier

Responsable : Equipe produit

Bénéfices : 1

Priorité : 3

33.1 Description :

Il est nécessaire d'avoir une vision globale des impacts du service numérique, à chaque phase de son cycle de vie afin : - d'identifier les leviers d'amélioration - de pouvoir mesurer le gain environnemental obtenu par la mise en œuvre des bonnes pratiques d'écoconception - d'éviter les transferts d'impacts

Les indicateurs d'impacts environnementaux les plus courants sont : - la consommation électrique - les émissions de gaz à effet de serre (GES) - la consommation d'eau

33.2 Mise en œuvre :

En tant qu'équipe produit, j'utilise l'outil de mesure de l'empreinte environnementale préconisé. Cet outil alimente un dashboard qui me permet de suivre l'impact environnemental du service tout au long de son cycle de vie.

33.3 Tests d'acceptance :

Un dashboard des impacts environnementaux existe et est alimenté tout au long du cycle de vie du service numérique.

34 ST-11 - Suivre les métriques fournies par les Core Web Vitals

Catégorie : - Stratégie - Pilotage

Phase : Cycle de vie entier

Responsable : Développeur + Référencier SEO

Bénéfices : 2

Priorité : 3

34.1 Description :

Les métriques de performance et de consommation d'un service numérique permettent d'identifier au plus tôt les problèmes (lenteur, surconsommation, etc.) liés à une fonctionnalité. En ayant connaissance de ces métriques, les équipes sont plus à même de trouver des solutions en amont de la mise en production et s'évitent ainsi des corrections fastidieuses et coûteuses. Les Core Web Vitals, ou Signaux Web Essentiels, sont 3 indicateurs définis par Google et mesurant la performance d'une page en fonction de données d'utilisation réelles : - LCP (Largest Contentful Paint) : temps nécessaire pour afficher le plus grand élément visible. - FID (First Input Delay) : temps nécessaire pour que la page réponde aux actions des utilisateurs. - CLS (Cumulative Layout Shift) : fréquence à laquelle les utilisateurs subissent des changements de mise en page inattendus.

Les standards définis par Google sont les suivants :

	Bon	Amélioration nécessaire	Médiocre
LCP	< 2,5s	<= 4s	> 4s
FID	<100ms	<= 300 ms	> 300 ms
CLS	< ?	<= 0,25	> 0,25

Pour en savoir plus, se référer à la documentation Google associée ¹.

34.2 Mise en œuvre :

En tant que développeur, plusieurs outils sont à ma disposition pour surveiller les Core Web Vitals, par exemple la Google Search Console ² ou encore l'outil WebPageTest ³. Cet outil alimente un dashboard qui me permet de suivre les Core Web Vitals du service tout au long de son cycle de vie.

34.3 Tests d'acceptance :

- Les Core Web Vitals sont suivis tout au long du cycle de vie du service numérique
- Le LCP est inférieur ou égal à 4 secondes, sinon, une alerte est remontée
- Le FID est inférieur ou égal à 300 millisecondes, sinon, une alerte est remontée
- Le CLS est inférieur ou égal à 0,25, sinon, une alerte est remontée

1. https://support.google.com/webmasters/answer/9205520?#status_explanation

2. <https://search.google.com/search-console/about>

3. <https://www.webpagetest.org/>

35 ST-12 - Mesurer l'expérience utilisateur

Catégorie : - Stratégie - Pilotage

Phase : Cycle de vie entier

Responsable : UX/UI Designer

Bénéfices : 2

Priorité : 3

35.1 Description :

L'UX a pour objectif d'offrir une expérience utilisateur fluide, agréable et intuitive. Les principes d'éco-conception et de l'UX se rejoignent donc sur plusieurs points : l'utilisabilité, la simplicité et l'efficacité, afin de répondre au besoin de l'utilisateur le plus rapidement et le plus efficacement possible. Ainsi, les indicateurs mesurant l'expérience utilisateur reflètent aussi le niveau d'écoconception d'un service numérique.

35.2 Mise en œuvre :

En tant qu'UX/UI Designer, je surveille a minima : - TTI (Time To Interactive) : temps nécessaire à une page pour devenir interactive pour l'utilisateur, c'est-à-dire pour qu'elle réagisse au clic - FCP (First Contentful Paint) : moment où le tout premier élément défini dans le DOM est rendu par le navigateur - Temps moyen passé par un utilisateur sur un parcours pour en estimer sa complexité - Distribution des clics : Part de l'ensemble des clics sur les éléments d'une page. La distribution des clics est notamment utile pour déterminer si un élément est utile ou non. - Taux d'abandon : le nombre d'utilisateurs ayant quitté un processus avant sa finalisation. Cet indicateur permet de cibler les parcours et éléments moins performants.

Des outils d'analytics tels que Google Analytics et des outils de qualité de page comme Lighthouse me permettent de suivre ces indicateurs.

35.3 Tests d'acceptance :

- Les indicateurs d'UX sont vérifiés tout au long du cycle de vie du service numérique
- Le TTI est inférieur à 7,3s
- Le FCP est inférieur à 4s
- Le temps moyen passé par l'utilisateur sur un parcours n'a pas varié de manière anormale depuis la dernière modification
- La distribution des clics ne laisse pas entrevoir des éléments inutilisés
- Le taux d'abandon n'a pas subi une forte augmentation depuis la dernière modification

36 ST-13 -Être attentif aux besoins des utilisateurs et vérifier l'apport de valeur du produit

Catégorie : - Stratégie - Pilotage

Phase : Run

Responsable : PO/AMOA

Bénéfices : 2

Priorité : 3

36.1 Description :

Lors de la phase d'utilisation, l'utilité de chaque fonctionnalité doit être réinterrogée au vu des usages et de leurs évolutions. Il s'agit ici de prendre du recul sur le service, et de valider l'intérêt réel et l'apport réel de chaque fonctionnalité dans le contexte actuel.

36.2 Mise en œuvre :

En tant que Product Owner ou AMOA, je mesure et j'analyse régulièrement : - La satisfaction des utilisateurs de mon service numérique (par exemple via des questionnaires, des indicateurs comme le NPS, etc.), la solution peut être définie lors de la session de travail sur les indicateurs - Les données d'analytics pour mieux comprendre le besoin des utilisateurs et leur comportement - L'utilisation des fonctionnalités (AT Internet) afin d'identifier les fonctionnalités qui tendent à être plus ou moins utilisées

36.3 Tests d'acceptance :

- La satisfaction des utilisateurs est mesurée tout au long du cycle de vie du service numérique

- Les données d'analytics sont surveillées tout au long du cycle de vie du service numérique
- L'utilisation des fonctionnalités est surveillée tout au long du cycle de vie du service numérique

37 ST-14 -Maîtriser la taille du DOM

Catégorie : - Stratégie - Pilotage

Phase : Cycle de vie entier

Responsable : Equipe Produit

Bénéfices : 2

Priorité : 3

37.1 Description :

Le DOM peut se définir de façon simplifiée comme un ensemble de balises HTML ou XML décrivant la structure de la page chargée par le navigateur. Plus le DOM est complexe, c'est-à-dire plus il comporte de balises, plus il faut de puissance (RAM et CPU) pour l'afficher. Ainsi, plus la puissance pour afficher la page est importante et plus cela contribue à raccourcir la durée de vie de l'ordinateur sur lequel il s'exécute et plus la consommation d'énergie associée à l'utilisation du service numérique est importante. Ces deux conséquences entraînent des impacts environnementaux non négligeables. Limiter la complexité de la page est donc la taille du DOM est donc primordial. A ce sujet, selon le guide des développeurs web de Google, un arbre de DOM optimal inclut les caractéristiques suivantes : - Moins de 1500 éléments au total (voire, si possible, moins de 800 éléments) - Une profondeur maximale de 32 éléments - N'a pas d'élément parent avec plus de 60 éléments enfants

37.2 Mise en œuvre :

En tant qu'équipe produit, je définis le nombre maximum d'éléments du DOM que ma page ne doit pas dépasser lors de la session de travail sur les indicateurs. En tant que développeur, j'utilise un outil en capacité de m'informer sur le nombre d'éléments du DOM . L'application de certaines bonnes pratiques de ce référentiel contribuent à minimiser le nombre d'éléments du DOM (ex : supprimer les éléments décoratifs, utiliser des spritesheets CSS, combiner les fichiers javascript et CSS, etc.)

37.3 Tests d'acceptance :

- Un nombre d'éléments maximal par page (ex : 600 éléments maximum) a été défini en fonction des contraintes du service numérique
- Le nombre d'éléments du DOM est inférieur à la limite définie pour chaque page

38 ST-15- Limiter le nombre de requêtes

Catégorie : - Stratégie - Pilotage

Phase : Cycle de vie entier

Responsable : Equipe Produit

Bénéfices : 2

Priorité : 3

38.1 Description :

Lorsque le navigateur affiche une page, il doit émettre une requête vers un serveur HTTP pour chaque fichier à télécharger. Plus le nombre de requêtes est important, plus les mécanismes de traitements et d'échanges vont consommer des ressources et donc augmenter les impacts environnementaux associés. Dès lors, il convient de limiter au maximum le nombre de requêtes.

38.2 Mise en œuvre :

En tant qu'équipe produit, je définis un nombre de requêtes maximum par écran lors de la session de travail sur les indicateurs et suis ce nombre avec l'outil préconisé (cf page 11). Le nombre de requêtes doit tenir compte de toutes les ressources téléchargées (composants d'interface, données, contenus, scripts, feuilles de style...).

38.3 Tests d'acceptance :

- Un nombre de requêtes maximal par écran (ex : 40 requêtes maximum) a été défini en fonction des contraintes du service numérique
- Le nombre de requêtes par écran est inférieur à la limite définie

39 ST-16 - Établir et respecter un budget environnemental

Catégorie : - Stratégie - Pilotage

Phase : Cycle de vie entier

Responsable : Equipe Produit

Bénéfices : 1

Priorité : 3

39.1 Description :

De la même façon qu'un projet numérique est contraint par un budget financier ou un planning, il pourrait être contraint par un budget environnemental. Par exemple, des contraintes sur le poids des pages pourraient être imposées (en Mo).

39.2 Mise en œuvre :

En tant qu'équipe produit, lors de la session de travail sur les indicateurs, j'établis un budget environnemental. Les différentes étapes de cet atelier sont les suivantes : - Définir les conditions de connexion les plus difficiles pour l'utilisateur et pour lesquelles la DSI souhaite que le service numérique puisse fonctionner (exemple : mauvaise connexion 3G) - Décider le temps de chargement possible dans ces conditions (exemple : 6 secondes) - Une fois ces critères définis, l'outil PerformanceBudget¹ renvoie le poids de page cible à atteindre. Si l'équipe trouve ce poids trop contraignant, des ajustements peuvent être réalisés (augmentation maximum à hauteur de 25% du budget) - Pour obtenir un budget global pour le service numérique, il faut multiplier le budget obtenu pour une page par le nombre de pages du produit. On obtient ainsi un budget global qu'il est possible de répartir entre les différentes pages. Tout dépassement du budget doit être justifié (comparaison avec un site concurrent écoconçu, etc.)

1. <https://www.performancebudget.io/>

Par exemple, si je veux que chaque page du service puisse se charger en 6 secondes avec une mauvaise connexion 3G, l'outil PerformanceBudget préconise un budget de 573 ko par page. Sachant que la page d'accueil la plus légère du site concurrent est Indeed et que son poids est 1285 ko, soit plus de 2 fois le budget recommandé, je peux me permettre d'augmenter le budget préconisé, dans la limite des 25%. Admettons que je choisisse 25%, mon budget s'élève maintenant à 716 ko. Supposons que le service numérique soit composé de 10 pages. Le budget global du service numérique est 7,16 Mo. Je peux choisir de répartir ce budget entre les différentes pages de mon service numérique. Par exemple; j'alloue davantage de budget à ma page d'accueil (ex : 1 Mo), qui est souvent plus lourde que les autres pages.

39.3 Tests d'acceptance :

- Un budget environnemental doit être défini et suivi avec l'outil préconisé (cf page 11)
- Chaque page créée doit respecter le budget environnemental défini

40 ST-17 - Vérifier la compatibilité du service numérique avec des faibles débits

Catégorie : - Stratégie - Pilotage

Phase : Build

Responsable : Equipe Produit

Bénéfices : 2

Priorité : Optionnel

40.1 Description :

Lorsqu'un service numérique s'adresse à un large public, il est impossible de maîtriser le niveau de connectivité. Cependant, il est essentiel de veiller à ne pas exclure certains publics qui n'ont pas accès à des hauts débits (débit inférieur à 512 kbit/s, équivalent à de la 3G). Dès lors, il est nécessaire d'avoir un service numérique nécessitant peu de débit pour éviter les fractures numériques. Minimiser le débit nécessaire au fonctionnement d'un service numérique sera aussi bénéfique pour l'environnement, car moins de données seront transférées sur le réseau.

40.2 Mise en œuvre :

En tant qu'équipe produit, je définis les pires conditions dans lesquelles mon service numérique doit pouvoir fonctionner (2G/3G, sur un terminal ancienne génération, etc.). En phase de conception et en phase de recette d'un service numérique, je teste les fonctionnalités critiques du service en simulant ces conditions dégradées.

40.3 Tests d'acceptance :

- Les conditions dégradées dans lesquelles le service numérique doit pouvoir fonctionner ont été définies
- Tout au long du cycle de vie du service numérique, des tests sont réalisés sur les fonctionnalités essentielles du service numérique pour déterminer si elles sont réalisables dans des conditions dégradées

41 UX-01 - Faire en sorte que la page s'adapte à tous les terminaux d'affichage

Catégorie : - Principes de design - UX

Phase : Build

Responsable : Equipe Produit

Bénéfices : 1

Priorité : 3

41.1 Description :

Le service numérique qui ne peut pas fonctionner sur des terminaux aux dimensions d'écrans variées peut pousser l'utilisateur à l'achat d'autres terminaux. Or, 80% des impacts environnementaux du Numérique sont induits par la fabrication du matériel informatique. Dès lors, il paraît essentiel de limiter l'achat et donc la fabrication de nouveaux terminaux. Rendre le service numérique accessible via des terminaux aux dimensions d'écrans variées est un levier important pour y remédier.

41.2 Mise en œuvre :

- En tant que designer, lorsque je réalise la maquette de ma page, je prends en compte l'aspect responsive.
- En tant que développeur et/ou intégrateur, je traduis cet aspect responsive dans le code que je produis (cf fiche pratique [Proposer plusieurs tailles d'images]).

41.3 Tests d'acceptance :

Les fonctionnalités critiques du service numérique s'affichent correctement sur différentes tailles d'affichage (ordinateur de bureau, tablette et mobile)

42 UX-02 - Adopter une approche mobile first si cela est pertinent

Catégorie : - Principes de design - UX

Phase : Build

Responsable : PO et UX/UI designer

Bénéfices : 1

Priorité : 2

42.1 Description :

L'approche mobile first consiste à concevoir un service numérique pour les terminaux mobiles, et élargir sa couverture fonctionnelle pour de plus grands écrans uniquement si l'apport fonctionnel/ergonomique est justifié. Les avantages d'une approche mobile first sont : - D'aller à l'essentiel et ainsi de réduire les fonctionnalités et contenus accessoires - De s'assurer que le service fonctionnera correctement sur les appareils mobiles

Minimiser le nombre de fonctionnalités développées permettra d'éviter la consommation induite par l'utilisation de fonctionnalités accessoires et donc peu ou pas utilisées et donc l'empreinte environnementale associée.

42.2 Mise en œuvre :

- En tant que Product Owner, je développe la stratégie du service numérique et priorise ses fonctionnalités en prenant en compte les enjeux « mobile first ».
- En tant que UX/UI Designer, à la place de concevoir une maquette adaptée en priorité pour un desktop et ensuite la dégrader en version mobile, je réalise une maquette pour une version mobile et je l'adapte ou je l'enrichie (si besoin) pour une version desktop.

42.3 Tests d'acceptance :

Si le service numérique est utilisé à au moins 30% par des devices mobiles : - Il est conçu pour une version mobile (les maquette sont réalisées pour mobile, les usages sont pensés pour une navigation sur mobile et les tests sont réalisés sur version mobile). Des ajustements peuvent éventuellement être réalisés pour étendre l'affichage pour les desktops. - Lors de la navigation sur mobile, aucune page ne présente de défaut d'affichage

43 UX-03 - Fournir un moyen de contrôle à l'utilisateur sur les contenus et services

Catégorie : - Principes de design - UX

Phase : Build

Responsable : Equipe produit

Bénéfices : 1

Priorité : Optionnel

43.1 Description :

Selon les usages, le niveau de qualité des contenus ne nécessite pas d'être élevé, et certaines fonctionnalités peuvent parfois être inutiles. Ainsi, si de telles différences sont observées au niveau des usages, il peut être intéressant de permettre à l'utilisateur d'ajuster certains paramètres de l'interface selon son besoin afin de réduire son empreinte environnementale. Des exemples : - Des fonctions proposant de choisir la définition de l'image à télécharger, - Des possibilités de choix en résolutions des médias (vidéos, sons, images, documents), - Des possibilités de désactiver l'affichage des médias, etc.

En revanche, si tous les besoins en termes de contenus et fonctionnalités sont similaires, il ne sera pas recommandé de proposer plusieurs alternatives (cela enrichit l'interface en Javascript), mais bien de s'en tenir à l'alternative la plus sobre.

43.2 Mise en œuvre :

- En tant que PO ou UX/UI Designer, j'analyse les cas d'usages de mon service et si ces cas d'usages peuvent se traduire différemment au niveau de l'interface, et que cela semble pertinent, je définis comment les paramètres pourront être ajustés par l'utilisateur.
- En tant que développeur, je mets en œuvre la solution imaginée par le PO et UX/UI Designer.

43.3 Tests d'acceptance :

- La pertinence de mettre en place un système de contrôle des contenus et services pour l'utilisateur a été évaluée
- Si cela s'avère pertinent, ces moyens de contrôle ont été mis en œuvre sur l'interface

44 UX-04 - Bien connaître les différents segments utilisateurs et leurs problèmes

Catégorie : - Personas - UX

Phase : Build

Responsable : PO/AMOA

Bénéfices : 2

Priorité : 3

44.1 Description :

Pour répondre au plus juste aux besoins des utilisateurs du service numérique, il est indispensable d'en connaître ses cibles, leurs usages, leurs besoins et leurs comportements, afin de ne pas surcharger les applicatifs numériques en fonctionnalités et contenus, ni les appauvrir au point qu'ils ne répondent pas aux attentes.

44.2 Mise en œuvre :

En tant que Product Owner ou AMOA, plusieurs moyens me permettent de définir les segments utilisateurs : - Observer, certains personas sont facilement identifiables - Réaliser des entretiens ou des sondages avec les utilisateurs - Analyser les solutions déployées en interne - S'inspirer des cibles utilisatrices identifiées par les solutions concurrentes

J'approfondis leurs problèmes et leurs contextes d'utilisation lors d'entretiens dédiés pour proposer les meilleures solutions possibles et écarter les autres.

44.3 Tests d'acceptance :

Des documents liés à l'étude du besoin existent et sont disponibles, par exemple : - Entretiens utilisateurs - Benchmarks des solutions concurrentes - Autres documents (à définir selon le contexte)

45 UX-05 - Ne garder que les fonctionnalités essentielles

Catégorie : - Fonctionnalités - UX

Phase : Build

Responsable : PO et métier

Bénéfices : 2

Priorité : 3

45.1 Description :

Les étapes précédant la phase de développement sont celles qui ont le plus gros effet de levier en termes de réduction des impacts environnementaux et économiques, En effet, si une fonctionnalité n'est pas développée, alors elle ne nécessitera pas d'optimisations a posteriori.

45.2 Mise en œuvre :

Détecter une fonctionnalité non essentielle est possible au moment de l'analyse de l'expression du besoin. Une multitude de méthodes permettent de vérifier l'utilité d'une fonctionnalité en amont de son développement : - La méthode MoSCoW permet de prioriser les fonctionnalités (Must have, Should have, Could have, Won't have) - L'impact mapping (outil qui permet de visualiser les stratégies possibles pour atteindre un objectif) favorise les idées qui vont contribuer à l'objectif donné et exclut celles qui n'ont aucun impact - La règle des 3U (Utile, Utilisable, Utilisée) garantit que les fonctionnalités développées seront pertinentes pour l'utilisateur et seront faciles d'utilisation - Les tests avec les utilisateurs permettent d'évaluer les décisions de conception par rapport à un ensemble représentatif d'utilisateurs

45.3 Tests d'acceptance :

Le nombre de fonctionnalités dont l'utilité n'a pas été vérifiée avec un panel d'utilisateurs avant développement est égal à 0.

46 UX-06 - Implémenter les fonctionnalités de manière sobre

Catégorie : - Fonctionnalités - UX

Phase : Run

Responsable : Métier / Équipe produit

Bénéfices : 3

Priorité : 3

46.1 Description :

Une fonctionnalité peut être implémentée de différentes manières, plus ou moins sobres. A titre d'exemple, une fonctionnalité de partage sur les réseaux sociaux peut être implémentée de différentes manières : via un plugin ou sous la forme d'une image avec un lien cliquable par exemple. Ces deux solutions n'ont pas la même empreinte environnementale (le plugin étant plus consommateur) mais remplissent la même fonction. Aussi, il faudra préférer la version sobre d'une fonctionnalité afin de ne pas induire une consommation inutile de ressources et donc des impacts environnementaux qui auraient pu être évités facilement.

46.2 Mise en œuvre :

En tant qu'équipe produit, j'opte pour des choix d'implémentation des fonctionnalités sobres. Je me réfère au tableau dédié à la bonne pratique pour orienter mes choix.

46.3 Tests d'acceptance :

- Le tableau comparatif a été utilisé pour orienter les choix d'implémentation des fonctionnalités

- Les choix d'implémentation d'au moins 50% des fonctionnalités sont sobres (couleur verte ou jaune dans le tableau comparatif)

47 UX-07 - Indiquer les impacts environnementaux des fonctionnalités très consommatrices

Catégorie : - Fonctionnalités - UX

Phase : Build

Responsable : Equipe produit

Bénéfices : 1

Priorité : Optionnel

47.1 Description :

Chaque fonctionnalité d'un service numérique induit des impacts environnementaux plus ou moins importants. Lorsqu'une fonctionnalité induit une forte consommation de ressources, il peut être intéressant d'en avertir l'utilisateur, afin qu'il se demande si utiliser cette fonctionnalité est réellement utile avant de passer à l'action. Cela permettrait d'éviter de gaspiller des ressources inutilement, et donc réduire l'empreinte environnementale liée à la navigation.

47.2 Mise en œuvre :

En tant qu'éditeur de contenus, lorsqu'une fonctionnalité est très consommatrice, je peux avertir l'utilisateur de plusieurs manières : - Par un pictogramme - En affichant son poids (ko) - En affichant son impact environnemental (si l'on dispose de ces données), il est d'ailleurs possible de vulgariser une empreinte carbone à l'aide de la calculatrice CO2¹

1. <https://impactco2.fr/convertisseur>

47.3 Tests d'acceptance :

100% des fonctionnalités fortement consommatrices sont accompagnées d'une précision sur les impacts qu'elles engendrent

48 UX-08 - Concevoir un parcours utilisateur simple et efficace

Catégorie : - Parcours utilisateur - UX

Phase : Cycle de vie entier

Responsable : PO et UX/UI Designer

Bénéfices : 2

Priorité : 3

48.1 Description :

Un des leviers de l'écoconception est de monopoliser le moins longtemps possible le réseau et les serveurs pour une action donnée. Cela revient à minimiser le temps passé par l'utilisateur sur le service numérique pour une action donnée. Un parcours utilisateur simple et efficace permet à l'utilisateur du service numérique de réaliser rapidement l'action voulue et ce, sans effort, ce qui améliore l'expérience utilisateur et permet d'éviter des consommations énergétiques inutiles liées à un parcours long et frustrant.

48.2 Mise en œuvre :

L'équipe accompagnement UX peut assister les équipes dans la conception du juste produit au juste besoin utilisateur, notamment dans le cadre de processus complexes.

- Service numérique existant En tant que Product Owner, j'identifie les parcours utilisateurs à optimiser en priorité en fonction de leur trafic, des données d'analytics (pour identifier les points de friction notamment) ainsi que de l'empreinte environnementale qui leur est associée (calculée avec l'outil préconisé, cf slide 11). En tant que UX/UI Designer, je détermine avec le Product Owner les actions qui permettront d'optimiser l'usage de ces parcours : diminuer le nombre d'étapes, diminuer le nombre d'actions, supprimer l'inutile, identifier les cas d'échecs,

optimiser les temps de réponse... Je réalise les maquettes en conséquence de ces choix et en fait la proposition au Product Owner.

- Nouveau service numérique En tant que PO et UX/UI Designer, nous réalisons un parcours utilisateur nativement simple et efficace après avoir mené des réflexions pour minimiser le nombre d'étapes et d'actions à réaliser par l'utilisateur et en nous assurant que le parcours utilisateur permet effectivement de réaliser l'action prévue. *Des optimisations pourront être, à titre d'exemple, proposer une nouvelle commande sur la base du contenu de la précédente, pouvoir réaliser une action sans inscription, copier/coller son RIB directement plutôt que le télécharger puis le transférer, mettre en avant les champs ou les filtres les plus utilisés.*

48.3 Tests d'acceptance :

Pour un parcours utilisateur existant donné : - Une analyse a été menée pour en identifier les leviers d'amélioration - Les leviers d'amélioration identifiés ont été mis en place

Pour un nouveau parcours utilisateur donné, le parcours a été conçu de sorte à minimiser le nombre d'étapes et d'actions à réaliser. Idéalement, ce parcours ne dépasse pas 3 étapes.

49 UX-09 - Favoriser un parcours utilisateur au sein d'un seul onglet

Catégorie : - Parcours utilisateur - UX

Phase : Build

Responsable : PO et UX/UI Designer

Bénéfices : 2

Priorité : 1

49.1 Description :

L'ouverture d'une nouvelle fenêtre ou d'un nouvel onglet lors de la navigation complexifie le parcours utilisateur et peut désorienter l'utilisateur lorsqu'il n'y est pas préparé. L'utilisateur poursuit ainsi son parcours sur une nouvelle fenêtre ou un nouvel onglet, laissant le premier onglet inutilisé. Le navigateur mobilise des ressources système pour fonctionner, notamment la mémoire (RAM) et le processeur, et il en nécessite d'autant plus qu'il y a d'onglets ouverts (et ce, même s'ils sont inutilisés). Dès lors, il convient de limiter l'ouverture de nouveaux onglets ou fenêtres au cours d'un parcours utilisateur. Toutefois, certains navigateurs, comme Google Chrome, Microsoft Edge ou Mozilla Firefox, disposent d'une fonctionnalité qui permet d'automatiser la mise en veille des onglets restés inactifs pendant un certain temps. Lorsqu'un utilisateur revient aux onglets inactifs, le navigateur Web charge à nouveau ces onglets. La bonne pratique aura donc moins d'impact pour les personnes utilisant ces types de navigateurs.

49.2 Mise en œuvre :

En tant que PO et UX/UI Designer, nous faisons en sorte de construire un parcours utilisateur qui se réalise dans un seul et même onglet. Des exceptions peuvent être accordées selon le besoin utilisateur (page de recherche, actualités, sources, etc.).

49.3 Tests d'acceptance :

La navigation sur le service numérique ne déclenche pas l'ouverture automatique de nouvelles fenêtres/onglets sauf si cela est justifié.

50 UX-10 - Limiter l'ouverture de pop-ups

Catégorie : - Parcours utilisateur - UX

Phase : Build

Responsable : PO et UX/UI Designer

Bénéfices : 2

Priorité : 2

50.1 Description :

Une pop-up est une fenêtre qui s'ouvre devant la fenêtre principale sans que l'action soit à l'initiative de l'internaute. D'une part, une pop-up perturbe la navigation de l'internaute, allongeant ainsi la durée de son parcours utilisateur, ce qui va à l'encontre du principe de frugalité de l'écoconception et pose problème au niveau de l'accessibilité. D'autre part, les pop-ups alourdissent le poids des pages. L'allongement de la durée d'un parcours utilisateur et l'alourdissement des pages associées induit une consommation énergétique supplémentaire et donc une empreinte environnementale plus forte.

50.2 Mise en œuvre :

En tant que PO et UX/UI Designer, nous n'intégrons pas des pop-ups non essentielles à la bonne réalisation du parcours utilisateur.

50.3 Tests d'acceptance :

Lors de la navigation sur le service numérique, aucune pop-up non essentielle ne s'ouvre.

51 UX-11 - Implémenter un moteur de recherche interne

Catégorie : - Parcours utilisateur - UX

Phase : Build

Responsable : PO et UX/UI Designer

Bénéfices : 2

Priorité : 3

51.1 Description :

Le moteur de recherche est, avec le plan du site, les menus et le lien de retour à l'accueil, un des moyens fondamentaux pour réorienter et faciliter l'accès à l'information à l'utilisateur final. La présence d'un moteur de recherche sur un service numérique permet de diminuer l'impact énergétique lié à sa consultation en optimisant le temps nécessaire à la réalisation d'une action par l'utilisateur.

51.2 Mise en œuvre :

En tant que PO et UX/UI Designer, je propose une fonctionnalité de recherche simple intégrée au service numérique, soit : - En se basant sur une équipe appartenant à l'entreprise pour la développer - En utilisant le moteur de recherche intégré au CMS - En utilisant le moteur de recherche d'un éditeur tiers (moins recommandé dans le cadre de l'écoconception)

51.3 Tests d'acceptance :

Une fonctionnalité de recherche est présente sur le service numérique.

52 UX-12 - Permettre à l'utilisateur de relancer une recherche depuis sa page de résultats

Catégorie : - Parcours utilisateur - UX

Phase : Build

Responsable : UX/UI Designer

Bénéfices : 2

Priorité : 1

52.1 Description :

Utiliser un moteur de recherche ne donne pas forcément à tous les coups une réponse correcte et adaptée. Très souvent les utilisateurs reformulent leur recherche ou changent certains paramètres. La présence d'un lien permettant de relancer voire même de modifier ses paramètres permet d'améliorer et d'accélérer les opérations de recherche en évitant les retours à la page de recherche initiale, réduisant ainsi les consommations de ressources et donc les impacts environnementaux associés.

52.2 Mise en œuvre :

En tant qu'UX/UI Designer, dans chaque page de résultat de recherche, j'affiche le formulaire de recherche avec la saisie précédente de l'utilisateur, de manière à ce qu'il puisse modifier celle-ci et valider immédiatement une nouvelle recherche.

52.3 Tests d'acceptance :

Pour chaque page de résultat de recherche, l'utilisateur peut valider immédiatement une nouvelle recherche à l'aide du formulaire de recherche pré-rempli avec sa saisie précédente, modifiable.

53 UX-13 - Indiquer au début d'un parcours utilisateur les données et documents qui seront demandés

Catégorie : - Parcours utilisateur - UX

Phase : Build

Responsable : UX/UI Designer

Bénéfices : 2

Priorité : 1

53.1 Description :

Si au cours d'un parcours utilisateur il est demandé à l'utilisateur de fournir des documents et des données spécifiques et qui pourraient lui prendre du temps à retrouver, il conviendra de les indiquer au début du parcours. Ainsi, cela permettra : - d'augmenter le taux de réussite du processus - de permettre à l'utilisateur de préparer les éventuels documents nécessaires

Ainsi, cela évitera à l'utilisateur de consommer des ressources et de l'énergie pour afficher les différentes étapes du parcours (et donc d'induire des impacts environnementaux dans le même temps) qui ne seront finalement pas fructueuses.

53.2 Mise en œuvre :

En tant qu'UX/UI Designer, pour tout parcours utilisateur où l'utilisateur doit fournir des documents et données spécifiques, je prévois avant la première étape de celui-ci un espace où sera spécifiée la liste des documents ou informations qui devront être fournis par l'utilisateur.

53.3 Tests d'acceptance :

Pour chaque processus complexe, la liste des documents ou informations spécifiques nécessaires pour finaliser le processus est indiquée à l'utilisateur avant la première étape de celui-ci.

54 UX-14 - Indiquer à l'utilisateur lorsqu'un traitement en arrière plan est en cours

Catégorie : - Parcours utilisateur - UX

Phase : Build

Responsable : Equipe produit

Bénéfices : 1

Priorité : Optionnel

54.1 Description :

Parfois, lorsqu'un utilisateur clique sur un bouton, aucun changement d'interface ne se produit et rien ne lui indique que son action a été prise en compte. Si la génération de la réponse est longue, alors cela peut conduire l'utilisateur à cliquer plusieurs fois sur le bouton, induisant ainsi des requêtes supplémentaire (et donc une empreinte environnementale plus importante). Dès lors, il conviendra de rendre indisponible l'action qui génère le traitement (par exemple un bouton de soumission de formulaire) et informer l'utilisateur qu'un traitement est en cours.

54.2 Mise en œuvre :

En tant qu'équipe produit (UX/UI ou développeur), pour toute action d'arrière plan de longue durée, je prévois d'informer l'utilisateur que son action a bien été prise en compte et je l'invite à patienter grâce à différents moyens : - En affichant un message pour confirmer la prise en compte de sa demande et en l'invitant à patienter (en indiquant éventuellement une durée approximative de traitement) - En « grisant » le bouton cliqué précédemment pour empêcher à l'utilisateur de cliquer à nouveau sur ce même bouton

54.3 Tests d'acceptance :

Pour chaque traitement en arrière plan, vérifier que l'utilisateur est invité à patienter et qu'on ne puisse pas reproduire l'action tant que le traitement n'est pas terminé.

55 UX-15 - Minimiser les notifications

Catégorie : - Messages d'informations - UX

Phase : Build

Responsable : PO et métier

Bénéfices : 2

Priorité : 2

55.1 Description :

Les notifications permettent à un utilisateur d'être informé d'une nouvelle activité vis-à-vis d'un de ses besoins ou centres d'intérêts. Cependant, si les notifications sont utilisées de manière exacerbée (fréquence élevée, informations peu pertinentes, etc.), elles peuvent attirer inutilement l'attention de l'utilisateur, déclenchant ainsi des actions de sa part et donc une consommation de ressources informatiques supplémentaires. A titre d'exemple, en 2021, les médias français envoyaient en moyenne une dizaine de notifications par jour.

De plus, augmenter la fréquence des notifications n'accroît pas nécessairement la valeur du service numérique. En effet, la raison principale (71% des personnes ayant répondu à l'étude d'Appiterate Survey) pour laquelle les utilisateurs désinstallent une application mobile est que les notifications leur sont désagréables.

Dans ce contexte, il conviendra de privilégier la qualité d'une notification à la quantité.

55.2 Mise en œuvre :

En tant que Product Owner, je travaille avec le métier pour déterminer comment réduire au maximum les notifications envoyées via mon produit. Ainsi, si un système de notifications est mis en place dans mon produit, je réalise un atelier avec le métier, pour m'assurer que : - Des notifications ne sont envoyées que lorsque l'information est réellement pertinente.

Exemples de notifications pertinentes : rappel de rendez-vous, rappel d'entretiens, notification de confirmation, ou encore une notification à la demande de l'utilisateur pour être informé d'un sujet.

- La fréquence d'envoi des notifications est minimisée Pour définir la fréquence des notifications, il est essentiel de comprendre ses segments utilisateurs, leur mode de vie et leurs besoins. Ainsi, la fréquence d'une notification devra être ajustée en fonction du type de notification (e-mail, push, etc.) ainsi que de son importance (par exemple, une notification demandée par l'utilisateur pour l'alerter d'une offre d'emploi sera plus prioritaire qu'une notification sur les dernières actualités). D'une manière générale, il faut éviter d'envoyer plusieurs notifications dans une période courte. Les moments où les utilisateurs ont peu de probabilité de consulter leur terminal (ex: entre minuit et 6h du matin) sont aussi à éviter.

Lors de cet atelier, j'adopte une approche « user centrix » en vérifiant que les notifications apportent réellement de la valeur et sont envoyées à une fréquence suffisante.

55.3 Tests d'acceptance :

- Si le service numérique dispose d'un système de notifications, un atelier a été organisé entre le Product Owner et le métier afin de déterminer comment les notifications peuvent être limitées.
- Le nombre de notifications envoyées par jour par le service numérique est inférieur à 3 (sauf s'il s'agit d'alertes que l'utilisateur a mis en place volontairement)

56 UX-16 - Permettre à l'utilisateur de régler les notifications

Catégorie : - Messages d'informations - UX

Phase : Build

Responsable : UX/UI Designer

Bénéfices : 2

Priorité : 2

56.1 Description :

Toutes les notifications ne sont pas forcément pertinentes pour l'utilisateur et peuvent ainsi retenir inutilement son attention. Permettre à l'utilisateur de désactiver ou de régler la fréquence des notifications qu'il reçoit permet de s'assurer que seules les informations dont il a besoin lui sont transmises lorsqu'il le souhaite.

56.2 Mise en œuvre :

Mettre en place une solution de notifications que l'utilisateur peut facilement désactiver ou en modifier la fréquence.

56.3 Tests d'acceptance :

- Le menu de gestion des notifications est facile d'accès
- L'utilisateur peut désactiver les notifications ou choisir la fréquence de réception de ces notifications

56.4 Mesures :

Mesures à réaliser avec l'équipe dédiée

57 UX-17 - Minimiser le poids des e-mails

Catégorie : - Messages d'informations - UX

Phase : Build

Responsable : Editeur de contenu

Bénéfices : 1

Priorité : 2

57.1 Description :

Les envois de mails génèrent une charge environnementale importante, notamment lorsque les mails contiennent des contenus riches. Dans le cas de diffusion en masse, ces impacts sont démultipliés. Le cas des newsletters est particulièrement sujet à cette explosion de trafic.

57.2 Mise en œuvre :

En tant qu'éditeur de contenus, différentes pratiques me permettent de diminuer le poids des e-mails que j'envois : - Je privilégie l'envoi de fichiers compressés, en basse définition - Je privilégie l'intégration d'un lien dans le mail à l'envoi d'une pièce jointe - J'évite d'intégrer des images au mail - Je simplifie au maximum la signature, éviter les logos ou les mettre en basse définition - Je ne joins pas les images et feuilles de style, mais laisse plutôt l'option à l'utilisateur de les télécharger - Si possible, j'utilise du texte brut à la place du format HTML

57.3 Tests d'acceptance :

Aucun des e-mails envoyés ne dépasse 1 Mo.

58 UX-18 - Choisir un service tiers écoconçu

Catégorie : - Service tiers - UX

Phase : Build

Responsable : Équipe produit

Bénéfices : 2

Priorité : 2

58.1 Description :

Les services tiers sont des services proposés par des fournisseurs externes. Il peut s'agir, par exemple des publicités, analytics, tracking, contenus (image, texte, vidéo, etc.), chat, fonctionnalités avancées, scripts d'A/B testing... 94 % des sites web contiennent des scripts tiers, alors qu'ils alourdissent les pages, induisant ainsi des impacts environnementaux non négligeables.

58.2 Mise en œuvre :

En tant qu'équipe produit, lorsque plusieurs alternatives sont possibles pour un même service tiers, je choisis le service tiers le moins consommateur. Les critères à comparer sont par exemple : - La mise en place d'une démarche d'écoconception par le fournisseur - L'empreinte environnementale du service tiers, communiquée par le fournisseur ou évaluée au moyen d'outils de mesure par l'équipe produit

58.3 Tests d'acceptance :

Une analyse comparative a été menée pour chaque service tiers utilisé.

59 UX-19 - Désactiver les services tiers non nécessaires sauf demande explicite de l'utilisateur

Catégorie : - Service tiers - UX

Phase : Build

Responsable : Équipe produit

Bénéfices : 2

Priorité : 2

59.1 Description :

Les services tiers sont des services proposés par des fournisseurs externes. Il peut s'agir, par exemple de publicités, analytics, tracking, contenus (image, texte, vidéo, etc.), chat, fonctionnalités avancées, scripts d'A/B testing...

Chaque service tiers supplémentaire alourdit le poids d'une page. Or, bien souvent, tous les services tiers ne sont pas nécessaires au bon fonctionnement d'une page. Ne charger des contenus non indispensables qu'à la demande explicite de l'utilisateur permet d'optimiser la consommation de ressources liée à la consultation d'une page et ainsi minimiser l'empreinte environnementale induite.

Ce critère rejoint une des obligations du RGPD sur la demande de consentement avant de charger du contenu tiers.

59.2 Mise en œuvre :

En tant qu'équipe produit, j'identifie les services tiers utilisés par le service numérique en consultant un outil d'analyse de site tel que Pingdom, GTmetrix, WebPageTest, ou encore Chrome Devtools.

Je détermine ensuite les services tiers non nécessaires au fonctionnement du service numérique en analysant les besoins des utilisateurs.

Enfin, je fais en sorte de ne charger que les services tiers indispensables aux utilisateurs et leur permet de déclencher le chargement des services tiers non essentiels lorsqu'ils le souhaitent.

59.3 Tests d'acceptance :

- Une analyse des services tiers utilisés par le service numérique a été réalisée
- Seuls les services tiers identifiés comme indispensables sont chargés
- L'utilisateur a la possibilité de charger les services tiers non essentiels s'il le souhaite

60 UX-20 - Permettre à l'utilisateur de refuser facilement les mécanismes de consentement

Catégorie : - Service tiers - UX

Phase : Build

Responsable : Équipe produit

Bénéfices : 2

Priorité : 2

60.1 Description :

Les services tiers sont des services proposés par des fournisseurs externes. Il peut s'agir, par exemple des publicités, analytics, tracking, contenus (image, texte, vidéo, etc.), chat, fonctionnalités avancées, scripts d'A/B testing...

Chaque service tiers supplémentaire alourdit le poids d'une page. Ainsi, l'utilisateur doit minimiser les services tiers qu'il charge pour minimiser son empreinte environnementale. Or, il arrive que certains éditeurs de services numériques rendent ce processus compliqué et décourageant pour les utilisateurs, les forçant à télécharger plus de services tiers que ceux dont ils avaient l'intention de télécharger, et augmentant par la même occasion les impacts environnementaux induits.

Dès lors, charger le socle de services tiers minimal doit être une procédure facile et rapide pour l'utilisateur.

60.2 Mise en œuvre :

En tant qu'équipe produit, je mets en place un mécanisme de consentement facile à refuser pour l'utilisateur, par exemple : - J'affiche clairement un bouton permettant de refuser les cookies - Je minimise le nombre d'étapes permettant de refuser les cookies - Je ne mets pas en avant le bouton

« accepter les cookies » de sorte à attirer le regard de l'utilisateur dessus et le faire accepter les cookies sans qu'il en ait eu l'intention

60.3 Tests d'acceptance :

L'utilisateur peut refuser en une action simple le mécanisme de consentement.

61 UX-21 - Maintenir le sitemap à jour

Catégorie : - SEO - UX

Phase : Cycle de vie entier

Responsable : Référenceur SEO

Bénéfices : 1

Priorité : 2

61.1 Description :

Le sitemap (fichier XML qui se positionne à la racine du site) permet de faciliter l'indexation des pages stratégiques et des contenus d'un site web par les moteurs de recherche. Un sitemap contient au maximum 50 000 URLs qui sont autant de pages que l'on souhaite indexer (on peut cependant créer ce que l'on appelle un index de sitemap). Le sitemap est parcouru par le robot Google à chaque visite sur le site, il est donc nécessaire de le maintenir à jour dès lors que de nouvelles pages stratégiques sont livrées sur le site. Un sitemap non mis à jour peut contenir des URLs obsolètes qui consommeront des ressources auprès du robot Google et des serveurs de Pôle emploi à défaut de les consacrer aux pages stratégiques.

61.2 Mise en œuvre :

En tant que référenceur SEO, je fais en sorte que le sitemap soit mis à jour automatiquement en fonction des nouvelles URLs stratégiques.

Exemples : - <https://www.pole-emploi.fr/informations/sitemap.xml> : sitemap dynamique, qui se met automatiquement à jour dès lors qu'un nouvel article est contribué - <https://candidat.pole-emploi.fr/SiteMaps-LP1.xml> : sitemap statique, qui n'a pas vocation à être modifié car pas de projet de création de nouvelles pages M

61.3 Tests d'acceptance :

100% des pages stratégiques sont présentes dans le sitemap.

62 UI-01 - Utiliser un mécanisme de lazy loading

Catégorie : - Interface - UI

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

62.1 Description :

Le lazy loading consiste à charger et afficher les images situées sous la ligne de flottaison lorsque celles-ci entrent dans l'écran. Cette technique permet de ne pas charger les images tant que l'utilisateur ne les voit pas (ex: les images se situant plus bas sur la page). Ainsi, le lazy loading garantit que les images ne soient chargées que lorsque cela est nécessaire. Cela permet d'économiser de la puissance de traitement, et donc la consommation d'énergie associée étant donné que les utilisateurs ne scrollant pas ne téléchargent pas les images concernées.

62.2 Mise en œuvre :

En tant que développeur, j'ajoute l'attribut loading="lazy" sur les éléments ou <iframe> :

```
  
<iframe src="video-player.html" title="..." loading="lazy"> </iframe>
```

62.3 Tests d'acceptance :

L'attribut loading="lazy" est présent sur les contenus le nécessitant

63 UI-02 - Éviter le défilement de contenus

Catégorie : - Contenus - UI

Phase : Build

Responsable : UX/UI Designer

Bénéfices : 2

Priorité : 3

63.1 Description :

L'infinite scroll est une technique qui permet aux utilisateurs de faire défiler le contenu sans apercevoir la fin de la page sur un site internet. Lorsqu'elle est utilisée, cette technique entraîne une augmentation de l'utilisation (et donc la consommation) de la mémoire de l'utilisateur et donc du processeur, ce qui affecte directement l'impact environnemental associé.

63.2 Mise en œuvre :

En tant qu'UX/UI Designer, je propose une solution de pagination ou un bouton « voir plus » sur les pages présentant une liste dont le nombre de contenus est supérieur à 10.

63.3 Tests d'acceptance :

- Aucune page n'est constituée d'un scroll infini
- Une page/liste est limitée à 10 éléments maximum

64 UI-03 - Fournir des CSS print optimisés

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur

Bénéfices : 1

Priorité : 1

64.1 Description :

Les CSS print permettent de proposer une version imprimable d'une page web. Les pages web sont composées d'éléments (headers, footers, etc) qui ne sont pas forcément pertinents pour l'utilisateur et qui induiraient donc du gaspillage de ressources à l'impression. Des CSS print sont déjà proposés sur pole-emploi.fr, mais ils pourraient gagner en légèreté.

64.2 Mise en œuvre :

En tant que Product Owner, je sélectionne les pages pour lesquelles des CSS print doivent être mis à disposition (les pages importantes et destinées à être imprimées).

En tant que développeur ou intégrateur, je réalise un CSS print optimisé en utilisant : - Une police de caractères économe en encre (Century Gothic, par exemple) - Un fond blanc - Uniquement le contenu pertinent pour l'utilisateur. Ainsi, le header, le footer, le menu, le sidebar, etc. doivent être masqués. Toutes les images sauf celles du contenu doivent être supprimées.

Un exemple de CSS print optimisé :

```
body { background-color :#fff; font-family :Serif; font-size :15pt; }  
#page { margin :0; border :none; }  
#banner, #menuright, #footer { display :none; }
```

```
h1#top { margin :0; padding :0; text-indent :0; line-height :25pt; font-size :25pt; } (...)
```

64.3 Tests d'acceptance :

Les pages disposent d'un CSS print optimisé.

65 UI-04 - Minimiser le nombre de vidéos par page

Catégorie : - Contenus - UI

Phase : Build

Responsable : Editeur de contenus

Bénéfices : 2

Priorité : 3

65.1 Description :

Les flux vidéo génèrent un volume de données important. Ainsi, réduire le nombre de vidéos mises en ligne en les remplaçant par une alternative (images, schéma, texte descriptif, etc...) est un réel levier de réduction de l'empreinte environnementale d'un service numérique.

65.2 Mise en œuvre :

En tant qu'éditeur de contenus, je limiter au maximum le recours au format vidéo pour présenter du contenu. Par exemple, une vidéo peut être remplacée par des images, un schéma ou encore un texte explicatif.

65.3 Tests d'acceptance :

Il n'y a pas de vidéo sur la page.

66 UI-05 - Raccourcir la durée des vidéos

Catégorie : - Contenus - UI

Phase : Build

Responsable : Editeur de contenus

Bénéfices : 2

Priorité : 3

66.1 Description :

L'impact des flux vidéo est principalement lié à la durée de la vidéo, à la qualité des images et du son, ainsi qu'au nombre de diffusion. Réduire la durée d'une vidéo est l'un des leviers facile de mise en œuvre pour réduire son impact, d'autant plus que si la vidéo est longue, il y a de grandes chances pour que les utilisateurs ne la visionnent pas en entier.

En effet, le temps d'attention des utilisateurs est assez limité. Ainsi, il est recommandé de réaliser des vidéos courtes et synthétiques. La durée optimale d'une vidéo présentant un sujet de manière globale est estimée à moins de 2 minutes et idéalement 1 minute. Les vidéos proposant un contenu thématique ou explicatif, qui traitent de sujets plus complexes, peuvent quant à elles durer plus longtemps.

66.2 Mise en œuvre :

En tant qu'éditeur de contenus, je limite au maximum la durée d'une vidéos selon les règles suivantes : - Aller à l'essentiel dans le discours et le contenu de la vidéo - Découper la vidéo avec un logiciel dédié pour n'en garder que les extraits pertinents - Diviser la vidéo en plusieurs vidéos, qui pourront alimenter chacune une page différente - Augmenter la vitesse de la vidéo si son contenu le permet

66.3 Tests d'acceptance :

- La durée d'une vidéo de présentation ne dépasse pas 2 minutes

- La durée d'une vidéo proposant un contenu thématique ou explicatif ne dépasse pas 15 minutes

67 UI-06 - Optimiser les vidéos

Catégorie : - Contenus - UI

Phase : Build

Responsable : Editeur de contenus

Bénéfices : 2

Priorité : 3

67.1 Description :

La définition d'une image correspond au nombre de pixels qu'elle peut afficher. Le poids (en octets) d'une image se calcule en multipliant le nombre de pixels qui la composent par leur poids. Ainsi, plus le nombre de pixels est important, et donc plus la définition d'une image est haute, plus son poids est élevé. Or, chaque octet transféré ou stocké sollicite des terminaux et des infrastructures gourmandes en énergie. Une plus haute définition induit donc des impacts environnementaux plus importants.

La définition à choisir dépend de la taille de l'écran du terminal utilisateur et du contenu proposé. Une définition supérieure à 1080p n'est pas nécessaire : une définition de 720p offre un rendu suffisant sur la plupart des écrans. Ainsi, imposer une définition inférieure ou égale à 720p aux vidéos proposées sur les service numérique permettra de réduire l'empreinte environnementale liée à la consultation de celui-ci.

Le nombre d'images par seconde (framerate) et le nombre de bits par secondes (bitrate) influent aussi sur la taille et donc l'impact environnemental du fichier. La compression des vidéos permet d'ajuster ces deux paramètres. Une multitude d'outils est disponible sur le marché pour ajuster la définition des vidéos et les compresser, par exemple [veed.io](https://www.veed.io/fr-FR/outils/compresser-vid%C3%A9o)¹, [video2edit](https://www.video2edit.com/fr/compresser-video)², [clideo](https://clideo.com/fr/compress-video)³, etc...

1. <https://www.veed.io/fr-FR/outils/compresser-vid%C3%A9o>

2. <https://www.video2edit.com/fr/compresser-video>

3. <https://clideo.com/fr/compress-video>

67.2 Mise en œuvre :

En tant qu'éditeur de contenus, je limite au maximum la durée d'une vidéos selon les règles suivantes : - Aller à l'essentiel dans le discours et le contenu de la vidéo - Découper la vidéo avec un logiciel dédié pour n'en garder que les extraits pertinents - Diviser la vidéo en plusieurs vidéos, qui pourront alimenter chacune une page différente - Augmenter le vitesse de la vidéo si son contenu le permet

67.3 Tests d'acceptance :

- Le niveau de compression de la vidéo a été adapté en fonction de son usage
- La définition des vidéos ne dépasse pas 720p sauf exceptions

68 UI-07 - Arrêter la lecture des vidéos hors du cadre de visualisation

Catégorie : - Contenus - UI

Phase : Build

Responsable : Intégrateur + dév front

Bénéfices : 2

Priorité : 3

68.1 Description :

Les vidéos lancées par l'utilisateur peuvent au cours de la navigation ne plus être visibles sur la zone écran présentée à l'utilisateur. La poursuite de la diffusion de ces contenus devient inutile et entraîne une consommation de ressources (et donc un impact environnemental) non nécessaire, ils doivent donc être arrêtés.

68.2 Mise en œuvre :

1. **Ecrire une fonction qui récupère les vidéos et les stoppe si elles sont hors du cadre**

Code :

```
function checkScroll() {  
  var videos = document.getElementsByTagName("video");  
  
  function isInView(el) {  
    var rect = el.getBoundingClientRect();           // absolute position of vid  
    return !(rect.top > $(window).height() || rect.bottom < 0);  // visible?  
  }  
}
```

```
}  
for(const video of videos){  
    if(isInView(video)){  
        video.pause()  
    }  
}  
}
```

Cas d'une iframe Youtube : Ajouter un paramètre enablejsapi=1 à l'url source de l'iframe. Dans le JS, envoyer la commande pauseVideo lorsque l'iframe est hors du cadre : `iframeElement.contentWindow.postMessage('{ "event": "command", "func": "pauseVideo", "args": "" }', '*')`

2. Appeler cette fonction lors d'un scroll :

Exemple en javascript : `window.addEventListener('scroll', checkScroll, false);`

Exemple en jQuery : `$(window).on('scroll', checkScroll);`

68.3 Tests d'acceptance :

- Le niveau de compression de la vidéo a été adapté en fonction de son usage
- La définition des vidéos ne dépasse pas 720p sauf exceptions

69 UI-08 - Désactiver l'autoplay des vidéos

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur

Bénéfices : 3

Priorité : 3

69.1 Description :

Les contenus actifs mobilisent d'importantes ressources techniques pour fonctionner, en particulier les vidéos. En effet, les usages vidéos représentent 80% du flux mondial de données. L'activité de ces composants doit être lancée à la demande de l'utilisateur pour réduire la consommation d'énergie et donc d'éviter des impacts environnementaux inutiles.

69.2 Mise en œuvre :

En tant que développeur, lorsque je veux désactiver la lecture automatique des vidéos, plusieurs situations se présentent à moi : - Intégration d'une vidéo Youtube ou Vimeo Dans le code `<iframe>` d'une vidéo Youtube ou Vimeo, la lecture automatique est désactivée par défaut. Je n'ai donc rien à faire. Pour modifier une vidéo dont la lecture automatique a été activée, je supprime le paramètre optionnel autoplay.

- Intégration d'une vidéo importée de l'ordinateur ou hébergée sur un autre site web L'élément HTML `<video>` intègre un contenu vidéo dans un document. Pour désactiver la lecture automatique d'une vidéo, je n'utilise pas l'attribut autoplay dans le code HTML, ou le supprime si il est déjà présent.

69.3 Tests d'acceptance :

- L'attribut autoplay n'est pas présent dans les balises <video>
- L'attribut autoplay n'est pas présent ou prend la valeur 0 dans les balises <iframe>

70 UI-09 - Ne pas jouer les vidéos en boucle

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur

Bénéfices : 3

Priorité : 3

70.1 Description :

Les contenus actifs nécessitent des ressources techniques, et donc une consommation d'énergie pour fonctionner. Les clips musicaux ont tendance à être lus à plusieurs reprises par les utilisateurs. C'est rarement le cas pour les contenus informatifs. Afin d'éviter un gaspillage d'énergie et donc induire des impacts environnementaux supplémentaires sans aucune finalité, une vidéo ne doit pas recommencer au début une fois que sa lecture est terminée.

70.2 Mise en œuvre :

En tant que développeur, lorsque je veux désactiver la lecture en boucle des vidéos, plusieurs situations se présentent à moi : - Intégration d'une vidéo Youtube Dans le code `<iframe>` d'une vidéo Youtube, la lecture en boucle est désactivée par défaut. Pour modifier une vidéo dont la lecture en boucle a été activée, je supprime le paramètre optionnel `loop`.

- Intégration d'une vidéo hébergée sur un autre site web ou sur pole-emploi.tv L'élément HTML `<video>` intègre un contenu vidéo dans un document. Pour désactiver la lecture en boucle d'une vidéo, je n'utilise pas l'attribut `loop` dans le code HTML, ou je le supprime si il est déjà présent.

70.3 Tests d'acceptance :

L'attribut loop n'est pas utilisé.

71 UI-10 - Prévoir une alternative texte aux vidéos

Catégorie : - Contenus - UI

Phase : Build

Responsable : UX/UI Designer + Editeur de contenu

Bénéfices : 2

Priorité : 1

71.1 Description :

Le texte, même mis en forme en HTML/CSS, utilise beaucoup moins de bande passante qu'une vidéo. Fournir aux utilisateurs une alternative textuelle à une vidéo leur permet s'ils le souhaitent de lire plutôt que de visionner, et donc de transférer moins de données. Si cette alternative textuelle a elle même une taille importante, elle peut ne pas être chargée par défaut mais suite à une action utilisateur.

Cette pratique est également bénéfique pour l'accessibilité : les mal entendants pourront lire le contenu et y auront donc accès, de même pour les mal voyants, si le texte inclut une description des éléments des vidéos qui ne sont que visibles.

Cette pratique est également bénéfique pour le référencement, les moteurs de recherches pouvant plus facilement analyser le texte que la vidéo.

71.2 Mise en œuvre :

En tant qu'UX/UI Designer, je prévois sur la maquette une zone de texte à proximité de la vidéo pour accueillir la retranscription de la vidéo en texte.

En tant qu'éditeur de contenu, je retranscris le contenu de la vidéo en texte.

71.3 Tests d'acceptance :

Le nombre de vidéos sans alternative textuelle est inférieur ou égal à 50%.

72 UI-11 - Afficher la durée des contenus vidéo

Catégorie : - Contenus - UI

Phase : Build

Responsable : UX/UI Designer

Bénéfices : 2

Priorité : 1

72.1 Description :

Mentionner la durée du contenu vidéo lorsque l'utilisateur s'apprête à le télécharger ou à le consulter en ligne lui permet de choisir s'il veut regarder le contenu, et si c'est le cas, le moment où il souhaite le regarder. En effet : - l'utilisateur peut ne pas avoir assez de temps pour le regarder, et pourrait donc le télécharger ou consulter une nouvelle fois plus tard - l'utilisateur peut souhaiter une durée plus courte/longue et déciderait donc de ne pas regarder le contenu déjà téléchargé ou en cours de lecture - etc.

Ainsi, cela éviterait à l'utilisateur de télécharger ou de consulter une vidéo inutilement parce que sa durée ne lui convient pas. Cela permettrait par la même occasion d'éviter de mobiliser de l'énergie et des ressources inutilement, et donc d'éviter l'empreinte environnementale associée.

72.2 Mise en œuvre :

En tant qu'UX/UI Designer, je prévois un espace sur chaque maquette comportant un audio ou une vidéo pour y indiquer leur durée. Celle-ci peut être éventuellement indiquée de manière générique, ou sous forme d'un ordre de grandeur commun à un ensemble de contenus.

72.3 Tests d'acceptance :

Le nombre de contenus vidéo pour lesquels la durée n'est pas mentionnée est égal à 0

73 UI-12 - Minimiser le nombre d'audios par page

Catégorie : - Contenus - UI

Phase : Build

Responsable : Editeur de contenus

Bénéfices : 2

Priorité : 3

73.1 Description :

Les flux audio génèrent un volume de données important. Ainsi, réduire le nombre d'audios mis en ligne en les remplaçant par une alternative (images, schéma, texte descriptif, etc...) est un réel levier de réduction de l'empreinte environnementale d'un service numérique.

73.2 Mise en œuvre :

En tant qu'éditeur de contenus, je limite au maximum le recours au format audio pour présenter du contenu. Par exemple, un audio peut être remplacée par des images, un schéma ou encore un texte explicatif.

73.3 Tests d'acceptance :

Il n'y a pas d'audio sur la page.

74 UI-13 - Désactiver l'autoplay des audios

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur

Bénéfices : 3

Priorité : 3

74.1 Description :

Les contenus actifs mobilisent d'importantes ressources techniques pour fonctionner. L'activité de ces composants doit être lancée à la demande de l'utilisateur pour réduire la consommation d'énergie et donc éviter les impacts environnementaux inutiles.

74.2 Mise en œuvre :

L'élément HTML `<audio>` intègre un contenu audio dans un document.

En tant que développeur, je désactive la lecture automatique d'un audio en : - n'utilisant pas l'attribut `autoplay` lorsque j'écris le code HTML - supprimant l'attribut `autoplay` si il est déjà présent dans le code HTML

74.3 Tests d'acceptance :

Il n'y a pas d'attribut `<autoplay>` dans la balise `<audio>`.

75 UI-14 - Utiliser des formats d'audios optimisés

Catégorie : - Contenus - UI

Phase : Build

Responsable : Editeur de contenus

Bénéfices : 2

Priorité : 3

75.1 Description :

Les flux audio génèrent un volume de données important. Si l'utilisation d'un audio s'avère essentielle, il conviendra d'optimiser le fichier en question. Il existe une multitude de formats d'audios sur le marché (MP3, OGG, AAC, FLAC, AIFF, WAV, etc.), chacun ayant ses propres caractéristiques. Bien souvent, les formats MP3, OGG ou AAC permettent d'avoir des fichiers plus légers que les formats à haute résolution type FLAC, AIFF ou encore WAV.

75.2 Mise en œuvre :

En tant qu'éditeur de contenus, avant publication, je teste les différents formats d'audios au moyen d'un outil en ligne (ex : [convertio¹](https://convertio.co/fr/)) afin de déterminer le format qui sera le plus léger.

75.3 Tests d'acceptance :

Plusieurs formats ont été testés afin de déterminer le plus léger.

1. <https://convertio.co/fr/>

76 UI-15 - Ajuster le niveau de compression des audios

Catégorie : - Contenus - UI

Phase : Build

Responsable : Editeur de contenus

Bénéfices : 2

Priorité : 3

76.1 Description :

Les flux audio génèrent un volume de données important. Si l'utilisation d'un audio s'avère essentielle, il conviendra d'optimiser le fichier en question. En particulier, le bitrate (débit), le ratio (taux de compression) et la fréquence d'échantillonnage peuvent être ajustés.

76.2 Mise en œuvre :

En tant qu'éditeur de contenus, avant publication, j'utilise un outil d'optimisation (ex : hitpaw¹) ou d'édition d'audios et ajuste les paramètres afin d'obtenir le fichier le plus léger possible adapté au contexte d'écoute de l'utilisateur. Par ailleurs, j'enregistre mon audio en Stéréo s'il s'agit d'une musique et en Mono pour un dialogue.

76.3 Tests d'acceptance :

Un effort d'optimisation a été réalisé sur les fichiers audio de la page

1. <https://online.hitpaw.fr/online-audio-compressor.html>

77 UI-16 - Ajuster le niveau de compression des audios

Catégorie : - Contenus - UI

Phase : Build

Responsable : UX/UI Designer

Bénéfices : 2

Priorité : 1

77.1 Description :

Mentionner la durée du contenu audio lorsque l'utilisateur s'apprête à le télécharger ou à le consulter en ligne lui permet de choisir s'il veut regarder le contenu, et si c'est le cas, le moment où il souhaite le regarder.

En effet : - l'utilisateur peut ne pas avoir assez de temps pour le regarder, et pourrait donc le télécharger ou consulter une nouvelle fois plus tard - l'utilisateur peut souhaiter une durée plus courte/longue et déciderait donc de ne pas regarder le contenu déjà téléchargé ou en cours de lecture - etc.

Ainsi, cela éviterait à l'utilisateur de télécharger ou de lire un audio inutilement parce que sa durée ne lui convient pas. Cela permettrait par la même occasion d'éviter de mobiliser de l'énergie et des ressources inutilement, et donc d'éviter l'empreinte environnementale associée.

77.2 Mise en œuvre :

En tant qu'UX/UI Designer, je prévois un espace sur chaque maquette comportant un audio pour y indiquer leur durée. Celle-ci peut être éventuellement indiquée de manière générique, ou sous forme d'un ordre de grandeur commun à un ensemble de contenus.

77.3 Tests d'acceptance :

Le nombre de contenus audios pour lesquels la durée n'est pas mentionnée est égal à 0

78 UI-17 - Minimiser le nombre d'images

Catégorie : - Contenus - UI

Phase : Build

Responsable : Editeur de contenu, UX/UI Designer, PO

Bénéfices : 2

Priorité : 3

78.1 Description :

Chaque image d'une page entraîne une consommation de ressources et donc un impact environnemental supplémentaire. Dès lors, il convient de minimiser le nombre d'images par page.

Pour commencer, il conviendra de supprimer toutes les images décoratives, c'est-à-dire n'apportant pas de valeur à l'utilisateur.

Une solution peut aussi être de réutiliser les images. Afficher les mêmes images au sein d'une même page permet de réduire le nombre d'images téléchargées sur la même page. Afficher les mêmes images d'une page à l'autre permet de limiter le téléchargement d'images. En effet, une fois téléchargée, l'image est mise en cache. Si l'image apparaît de nouveau sur une page lors de la navigation, le navigateur pourra obtenir l'image directement dans le cache plutôt que sur le serveur d'origine, ce qui permet de ne pas la télécharger une deuxième fois.

78.2 Mise en œuvre :

La mesure de la valeur d'un élément peut être déterminée selon la règle des 3U (est-ce que l'élément est Utile, Utilisable, Utilisée ?). Ainsi, en tant qu'UX/UI Designer et Product Owner et avec le métier, nous appliquons la règle des 3U pour chaque élément que nous souhaitons ajouter à l'interface utilisateur. Si un élément ne répond pas à au moins un des trois critères des 3U, alors nous supprimons cet élément.

En tant qu'éditeur de contenus et UX/UI Designer, nous essayons au maximum de réutiliser les visuels entre les pages du service numérique auquel nous contribuons.

Par exemple : - Les pictogrammes peuvent être réutilisées sur les différentes pages - Les images illustrant les articles, actualités ou encore les éléments d'une recherche (s'il y en a, cf bonne pratique UI-29) doivent être génériques afin de pouvoir être recyclées pour d'autres sujets.

78.3 Tests d'acceptance :

- La règle des 3U a été utilisée pour vérifier que chaque élément de la page est utile, utilisé et utilisable
- Le nombre d'images décoratives par écran est inférieur ou égal à 3
- 100% des images téléchargées sont affichées

79 UI-18 - Préférer le CSS aux images pour les éléments d'interface

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur et intégrateur

Bénéfices : 2

Priorité : 3

79.1 Description :

Le poids d'une feuille de styles est bien plus faible qu'une image, surtout si la feuille de style est compressée. De plus, l'appel d'une feuille de styles ne génère qu'une seule requête HTTP, contre un grand nombre si l'on emploie beaucoup d'images (une requête HTTP pour chaque image). Le poids transféré et le nombre de requêtes influant fortement sur l'empreinte environnementale d'une page, il conviendra d'utiliser les propriétés CSS à la place d'images pour réaliser l'interface.

79.2 Mise en œuvre :

En tant que développeur/intégrateur, je maximise l'utilisation de propriétés CSS à la place d'utiliser des images (ex : arrières plans).

79.3 Tests d'acceptance :

Le nombre d'images qui auraient pu être remplacées par du CSS est égal à 0

80 UI-19 - Générer des spritesheets CSS lorsque le contexte le permet

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 1

80.1 Description :

Le principe de cette bonne pratique est de regrouper les images de petite taille (celles de l'interface du site, par exemple) dans une seule image de plus grande taille appelée spritesheet. Ce procédé réduit significativement le nombre de requêtes HTTP. Les spritesheets CSS sont déjà utilisés sur pole-emploi.fr, mais certaines optimisations peuvent encore être réalisées.

80.2 Mise en œuvre :

De nombreux services en ligne gratuits permettent de générer des spritesheets : - CSS Sprites Generator Tool ¹ - Spritegen ² - CSS Sprites ³

80.3 Tests d'acceptance :

Des spritesheets CSS sont générés lorsque le contexte le permet.

1. <https://www.toptal.com/developers/css/sprite-generator>

2. <https://spritegen.website-performance.org/>

3. <http://csssprites.org/>

81 UI-20 - Minimiser les dimensions des images

Catégorie : - Contenus - UI

Phase : Build

Responsable : UX/UI Designer

Bénéfices : 2

Priorité : 3

81.1 Description :

Les dimensions d'une image sont proportionnelles à son poids : plus une image est grande, plus son poids est important. Il conviendra alors de minimiser au maximum la taille des images du service numérique.

81.2 Mise en œuvre :

En tant qu'UX/UI Designer, lors de la réalisation de la maquette de la page, je choisis des dimensions d'images minimales.

81.3 Tests d'acceptance :

- Des tests ont été réalisés avec des visuels de différentes tailles, pour déterminer jusqu'à quelles dimensions il était possible de réduire les images sans altérer l'esthétique de la page
- Suite aux tests, les images aux dimensions les plus petites et n'altérant pas l'esthétique de la page ont été retenues pour le design

81.4 Mesures :

Un exemple pour l'image principale de la homepage de pole-emploi.fr :

Taille de l'image	Poids de l'image
1920x790	165 KB
1020x420	86.4 KB (-47%)

En diminuant les dimensions de l'image principale de la homepage de pole-emploi.fr de cette manière, 1,5 tonnes de CO₂eq* auraient pu être évitées au mois d'avril. De manière plus concrète, cela correspondrait à 7 777km en voiture ou encore 3 311 litres d'eau en bouteille.

**15 663 548 de visites sur la homepage de pole-emploi.fr en avril

82 UI-21 - Compresser et ajuster la résolution des images

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

82.1 Description :

Afin de réduire leur poids, les images peuvent être minifiées et compressées. - Minifier un fichier consiste à supprimer les commentaires, les espaces inutiles et les sauts de ligne. En effet, ces éléments, qui aident à rendre le code plus lisible, augmentent la taille du fichier alors qu'ils sont ignorés par le navigateur. Seuls les fichiers SVG sont concernés par la minification.

- Compresser les images permet de réduire leur poids. Il faut toutefois veiller à ajuster le taux de compression en fonction de l'image considérée afin de ne pas altérer sa qualité.

82.2 Mise en œuvre :

En tant que développeur, une multitude d'outils sont à ma disposition pour optimiser des images, par exemple : - Pour l'optimisation webp : l'outil cwebp - Pour l'optimisation jpeg : l'outil jpegoptim - Pour l'optimisation png : l'outil opting - Pour l'optimisation avif : l'outil avif - Pour l'optimisation gif : l'outil gifsicle - Pour l'optimisation svg : l'outil svgo

Ces outils disposent d'options qui permettent d'ajuster la compression des images.

82.3 Tests d'acceptance :

Des outils de compression ont été utilisés pour ajuster la compression des images La résolution des images est inférieure ou égale à 72 dpi (pour le vérifier, regarder les propriétés des images ou utiliser <https://pixelcalculator.com/fr>)

83 UI-22 - Fournir des formats d'images optimisés

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

83.1 Description :

Choisir le bon format d'images est crucial pour éviter de transporter des octets inutilement et ainsi économiser de la bande passante. Les formats .avif et .webp offrent de bien meilleures performances que les formats classiques, tels que .jpeg et .png, en revanche, ils ne sont pas pris en charge par tous les navigateurs. Le format .svg, quant à lui, est le format le plus adapté pour les éléments d'interface, les icônes ou les diagrammes.

83.2 Mise en œuvre :

En tant que développeur, dans la balise `<picture>` du code HTML, je déclare les images, dans l'ordre du plus léger au plus lourd en finissant par un format et une balise historique (par exemple : avif puis webp puis jpg).

Un exemple de trame :

```
<Picture>
  <source srcset="image.avif" type="image/avif">
  <source srcset="image.webp" type="image/webp">
  
</Picture>
```

Pour fabriquer les assets, j'utilise l'outil convert de imagemagick.

83.3 Tests d'acceptance :

- Le premier format proposé est le format avif, puis webp et enfin un format historique (png, jpg, gif, etc)
- Le premier format proposé est le format svg et enfin un format historique (png, jpg, gif, etc).

83.4 Mesures :

Nous avons testé la bonne pratique sur la homepage de pole-emploi.fr :

	Avec les formats d'origine	Après compression des images	Avec les images au format .webp	Avec les images au format .avif
Poids de la page	2,5 Mo	2,17 Mo (-13,2%)	1,95 Mo (-22%)	1,63 Mo (-34,8%)

Si l'on prend en compte les 15,6 millions de visites sur la homepage en avril 2022, on obtient : || Avec les formats d'origine | Après compression des images | Avec les images au format .webp | Avec les images au format .avif | |-----|-----|-----|-----|
|-----| || Poids de la page | 39,1 To | 33,9 To (-13,2%) | 30,5 To (-22%) | 25,5 To (-34,8%) |

Ainsi, en passant les images de la homepage de pole-emploi.fr au format .avif, on évite le téléchargement de 13,6 To sur le mois d'avril, ce qui équivaut à 16,6 tonnes de CO2eq évitées. Cela représente, concrètement, 86 010 km parcourus en voiture ou encore la fabrication de 663 smartphones.

84 UI-23 - Proposer plusieurs tailles d'images

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur et intégrateur

Bénéfices : 2

Priorité : 3

84.1 Description :

Les services numériques utilisent plusieurs tailles d'une même image en fonction de la taille de l'écran du terminal utilisateur. Il est préférable que les images soient disponibles dans les différentes tailles utilisées par le service et non qu'elles soient redimensionnées dans le code HTML avec les attributs height et width. En effet, le redimensionnement des images, lorsqu'il est effectué à la demande, génère des charges de traitements, des productions de fichiers intermédiaires qui introduisent des déperditions d'énergie à chaque demande.

84.2 Mise en œuvre :

En tant que développeur et/ou intégrateur, je propose plusieurs tailles pour les photos (non applicable pour les svg) en utilisant les attributs srcset et sizes.

Les attributs srcset et sizes permettent de fournir plusieurs images sources avec des indications pour aider le navigateur à faire le bon choix.

- srcset définit l'ensemble des images offertes au choix du navigateur, et la taille de chaque image
- sizes définit un ensemble de conditions pour le média (par ex. des largeurs d'écran) et indique quelle taille d'image serait la plus adaptée si certaines conditions sont satisfaites

Un exemple de trame :

```

```

84.3 Tests d'acceptance :

Deux tailles d'images sont proposées.

85 UI-24 - Limiter les animations Javascript/CSS

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur

Bénéfices : 3

Priorité : 2

85.1 Description :

Une animation est la visualisation d'un changement apportés à un objet ou un ensemble d'objets sur une période de temps donnée.

Les animations JavaScript/CSS peuvent être très coûteuses en termes de cycles CPU et de consommation mémoire. Elles déclenchent toutes une action de type (re)paint/(re)flow très coûteuse en ressources.

Par ailleurs, les animations sont généralement dépendantes de caractéristiques techniques des terminaux et génèrent des incompatibilités qui poussent artificiellement au renouvellement des équipements. Les animations détournent aussi l'attention des utilisateurs. Pour ces raisons, il convient de limiter au maximum les animations.

85.2 Mise en œuvre :

En tant que développeur, dans la mesure du possible, je n'utilise des animations que lorsqu'elles apportent réellement de la valeur aux utilisateurs.

Si une animation est nécessaire, se limiter aux propriétés CSS3 opacity et transform, et aux fonctions associées translate, rotate, scale de transform. Ces deux propriétés sont automatiquement optimisées par le navigateur, et l'animation peut être prise en charge par le processeur graphique (GPU).

En Javascript, au lieu d'utiliser `setTimeout()` et `setInterval()`, préférer la méthode native `requestAnimationFrame()` pour que le navigateur exécute l'animation au moment opportun. Cette méthode adapte la fréquence d'images au terminal de l'utilisateur.

Par ailleurs, chaque animation doit pouvoir être mise en pause par l'utilisateur.

85.3 Tests d'acceptance :

- L'apport de valeur de chaque animation a été démontré
- Chaque animation est implémentée de manière sobre (respect des guidelines du paragraphe « mise en œuvre »)

86 UI-25 - Limiter les carrousels automatiques

Catégorie : - Contenus - UI

Phase : Build

Responsable : UX/UI Designer

Bénéfices : 3

Priorité : 3

86.1 Description :

Un carrousel est un type de module qui fait la rotation du contenu d'une manière similaire à celle d'un diaporama, soit par une action de l'utilisateur ou par des transitions temporisées. Sa présence implique plusieurs conséquences : - Un alourdissement du poids des pages par la présence de CSS et JavaScript dédiés mais également par le contenu présent sur chaque écran du carrousel - Une complexité en termes d'assurance qualité web et d'accessibilité numérique - Une utilisation plus importante des ressources processeurs dans le cas d'un carrousel automatique

De plus, en réalité, les carrousels ne sont que très peu utilisés par les utilisateurs.

86.2 Mise en œuvre :

En tant qu'UX/UI Designer, je peux limiter le recours à un carrousel en : - Supprimant le carrousel - Remplaçant le carrousel, par exemple, par l'image la plus importante qu'il contient, ou un résumé des différents contenus qu'il présente - Si un carrousel est nécessaire, je limite le nombre d'écrans présents dans le carrousel, privilégie le lazy loading de ces images et n'autorise son déclenchement que suite à une action de l'utilisateur

86.3 Tests d'acceptance :

- Le nombre de carrousels sur la page est égal à 0

- Si la présence d'un carrousel s'avère nécessaire, il est optimisé (nombre d'écran inférieur ou égal à 2 et lazy loading activé)

87 UI-26 - Limiter le nombre de webfonts utilisées

Catégorie : - Contenus - UI

Phase : Build

Responsable : UX/UI Designer

Bénéfices : 2

Priorité : 3

87.1 Description :

Les webfonts sont des polices spécialement créées pour les sites Web. Elles sont généralement hébergées sur un serveur Web. Cela signifie que les polices Web sont téléchargées par le navigateur lors du rendu de la page Web, puis appliquées au texte. Chaque webfont propose un certain nombre de variantes (standard, light, semi-bold, bold, black...) et chaque variation est un nouveau fichier à charger. A l'inverse, une police système est une police déjà présente sur le système d'exploitation d'un terminal utilisateur et ne nécessite donc aucun téléchargement.

Afin de limiter le poids de la police utilisée, il faudra privilégier les polices systèmes et, si ce n'est pas possible, limiter le nombre de variantes de webfonts utilisées.

87.2 Mise en œuvre :

En tant que UX/UI Designer, plusieurs options s'offrent à moi pour limiter le nombre de webfonts utilisées : - Utiliser les polices systèmes Selon les systèmes d'exploitation, les polices systèmes peuvent être différentes. Pour s'assurer que le site s'affiche correctement pour tous les utilisateurs, il faut donc prévoir plusieurs polices.

En CSS, la propriété font-family permet de définir une liste, ordonnée par priorité, de polices à utiliser pour mettre en forme le texte de l'élément cible.

Un exemple de trame pour les fonts de type Monospace :

Font-family: SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace;

Un exemple de trame pour les fonts de type Sans Serif :

Font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Oxygen-Sans, Ubuntu, Cantarell, "Helvetica Neue", sans-serif;

- Réduire le nombre de webfonts utilisées Afin de réduire le nombre de fichiers téléchargés, utiliser la police personnalisée uniquement sur les titres et utiliser une police système pour le texte des paragraphes, par exemple. Cela permet de trouver un compromis entre personnalisation de la charte graphique et sobriété.

87.3 Tests d'acceptance :

Le nombre de fichiers de webfonts est inférieur ou égal à 2.

88 UI-27 - Utiliser les formats de police WOFF2 et WOFF

Catégorie : - Contenus - UI

Phase : Build

Responsable : UX/UI Designer

Bénéfices : 2

Priorité : 3

88.1 Description :

Le Web Open Font Format (WOFF) est un format de police de caractère comprimée pour un usage sur les sites web. Le format WOFF est supporté par quasiment tous les navigateurs. WOFF2 est une évolution du format WOFF, encore plus compressé, mais il n'est pas supporté par les anciennes versions de navigateurs. En CSS, la propriété `@font-face` permet l'utilisation de fontes personnalisées.

88.2 Mise en œuvre :

En CSS, la propriété `@font-face` permet l'utilisation de fontes personnalisées.

En tant que développeur, après avoir indiqué la fonte à utiliser, je liste les fontes de fallback (en cas d'indisponibilité d'une fonte, le navigateur tente d'utiliser sa suivante, etc).

Un exemple de trame :

```
@font-face {  
    font-family: "Open Sans";  
    src: url("/fonts/OpenSans-Regular-webfont.woff2") format("woff2"),  
         url("/fonts/OpenSans-Regular-webfont.woff") format("woff");  
}
```

88.3 Tests d'acceptance :

Le format WOFF2 puis le format WOFF sont proposés.

89 UI-28 - Optimiser les documents

Catégorie : - Contenus - UI

Phase : Build

Responsable : Créateur de contenus

Bénéfices : 2

Priorité : 3

89.1 Description :

Même si son interface a été « éco-conçue », le service numérique peut être alourdi par les contenus présentés. En particulier, de nombreuses optimisations peuvent être réalisées sur le contenu des documents qui y sont intégrés.

89.2 Mise en œuvre :

En tant que créateur de contenus, j'utilise les options d'export généralement proposées par les logiciels de traitement de texte ou de présentations pour réduire le poids des fichiers (compression des contenus et diminution de la résolution des images).

1. Compression

Plusieurs niveaux de compression sont généralement proposés. Tester les différents niveaux, du plus compressé au moins compressé, permet de choisir le rendu le plus approprié.

2. Résolution

Les résolutions suivantes sont conseillées : - Pour une visualisation en ligne : fournir un document comportant des visuels dont la résolution est inférieure ou égale à 72 dpi Pour une impression HD : fournir un document comportant des visuels en 150 ou 300 dpi

3. Autres optimisations

- Si possible, - Réduire la taille des images - Minimiser le nombre de polices téléchargées et leurs variantes
- Découper le document en chapitres, afin de limiter les téléchargements inutiles

89.3 Tests d'acceptance :

Le niveau de compression du document a été ajusté selon le contexte et de manière à minimiser le poids du document. D'autres pistes d'optimisations (cf mise en œuvre) ont été explorées et mises en place.

90 UI-29 - Remplacer la complétion automatique par une alternative induisant moins de requêtes

Catégorie : - Contenus - UI

Phase : Build

Responsable : PO et UX/UI Designer

Bénéfices : 1

Priorité : 2

90.1 Description :

L'autocomplétion est une fonctionnalité très répandue consistant à suggérer à l'utilisateur des résultats correspondant à sa recherche pendant sa saisie. Une requête au serveur va être envoyée à chaque caractère saisi pour récupérer les résultats correspondants. Or, bien souvent l'utilisateur sait ce qu'il veut rechercher. L'affichage des résultats est donc peu utile dans ces cas-là, d'autant plus qu'il émet plus de requêtes que nécessaire, ce qui induit des impacts environnementaux supplémentaires (cf fiche ST-13).

90.2 Mise en œuvre :

En tant que PO et UX/UI Designer, j'évite le recours à l'autocomplétion et privilégie plutôt les techniques d'aide existantes (prévention des erreurs, exemples, format de saisie, liste déroulante, etc.).

Si l'usage de l'autocomplétion reste pertinent dans le contexte du projet, il est également possible de limiter ses impacts :

- Ajouter un délai de quelques dixièmes de secondes entre l'ajout d'un caractère et la requête : cela permet de ne pas déclencher de requête si l'utilisateur n'a pas terminé sa saisie

- Limiter le nombre de résultats affichés par l'autocomplétion, priorisés par une note de pertinence
- Fixer un nombre de caractères minimal avant de chercher à compléter
- Si la taille de la base de données le permet, l'inclure dans le code html ou dans le local storage et effectuer l'autocomplétion côté client
- Mettre en cache les résultats des recherches avec pour clef la chaîne saisie pour moins solliciter la base de données

90.3 Tests d'acceptance :

Le nombre de champs en autocomplétion est inférieur à 20%

91 UI-30 - Limiter le poids et le format des fichiers que l'utilisateur doit téléverser

Catégorie : - Contenus - UI

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 1

91.1 Description :

Les interactions sur le front end peuvent nécessiter des transferts de fichiers entre le poste utilisateur et les serveurs de traitement. Les formats et volumes de fichiers acceptés par le service numérique doivent être indiqués à l'utilisateur pour éviter d'induire une consommation (liée au transfert, stockage et traitement des fichiers) alors que le processus sera voué à l'échec si les fichiers ne respectent pas les formats et volumes attendus. La limitation du volume des fichiers est indispensable pour réduire l'impact environnemental lié à leur transfert.

91.2 Mise en œuvre :

En tant que développeur, je peux mettre en œuvre deux actions de manière conjointe : - Indiquer à l'utilisateur la taille maximale et les formats acceptés par le service numérique - Limiter la taille et les formats des fichiers à transmettre au niveau du formulaire HTML

Dans un élément `<form>`, un élément `<input>` dont l'attribut `type` vaut `"file"` permet à un utilisateur de sélectionner un ou plusieurs fichiers depuis son appareil et de les uploader vers un serveur via un formulaire ou grâce à du code JavaScript via l'API File.

L'attribut `accept` permet de définir les formats de fichiers acceptés. Un exemple de trame :


```
<form>
    ...
    <input onchange="VerifyUploadSizeIsOK(this)" type="file" id="file-
upload"
    accept=".doc,.docx" />
    ...
</form>
```

La fonction Javascript suivante permet de limiter le poids des fichiers à transmettre :

```
<script type="text/javascript">
function VerifyUploadSizeIsOK()
{ var UploadFieldID = "file-upload";
  var MaxSizeInBytes = 2097152;
  var fld = document.getElementById(UploadFieldID);
  if( fld.files && fld.files.length == 1 && fld.files[0].size > MaxSizeInBytes
  {alert("The file size must be no more than " + parseInt(MaxSizeInBytes/1024/1024) + " Mo");
    fld.value = '';
    return false;}
  return true;} // function VerifyUploadSizeIsOK()
</script>
```

91.3 Tests d'acceptance :

- Le poids et le format des fichiers à transmettre par l'utilisateur sont précisés
- L'utilisateur ne peut pas envoyer de fichier dans un format différent et à un poids supérieur à ce qui est spécifié

92 UI-31 - Vérifier les formats attendus de la saisie de l'utilisateur avant soumission du formulaire

Catégorie : - Contenus - UI

Phase : Build

Responsable : PO/Développeur

Bénéfices : 1

Priorité : 2

92.1 Description :

Lorsqu'un utilisateur complète un formulaire, un processus de vérification des informations saisies (validation des données) peut être mis en place afin de vérifier que les champs obligatoires soient bien complétés ou encore que le format des données soit correct. Ce processus peut être effectué dans le navigateur, avant que les données n'aient été soumises au serveur (validation côté client).

La validation des données côté client permet d'éviter d'envoyer une requête inutile au serveur en cas d'erreur et donc d'économiser les impacts environnementaux associés. En effet, chaque requête repose sur des mécanismes de traitements et d'échanges, qui consomment de l'énergie.

92.2 Mise en œuvre :

En tant que développeur, je peux mettre en place la validation des données côté client de plusieurs manières : - la validation JavaScript, codée en JavaScript, entièrement personnalisable - la validation de formulaire intégrée avec les fonctions de validation de formulaire HTML5. Elle ne nécessite généralement pas de JavaScript, a de meilleures performances, mais elle n'est pas aussi personnalisable

92.3 Tests d'acceptance :

Un mécanisme de validation des données côté client a été mis en place pour 100% des formulaires du service numérique.

93 CO-01 - Concaténer les fichiers CSS

Catégorie : - Code - CSS

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 2

93.1 Description :

La combinaison de feuilles de style CSS permet de réduire le nombre de requêtes HTTP et donc les impacts environnementaux associés (cf fiche ST-13).

Cette bonne pratique ne sera plus valable si Pôle emploi adopte le protocole HTTP/2, qui concatène déjà les fichiers CSS.

93.2 Mise en œuvre :

En tant que développeur, j'utilise le plugin webpack-merge-and-include-globally de l'outil Webpack pour fusionner les fichiers CSS.

93.3 Tests d'acceptance :

Il y a au maximum 3 fichiers CSS.

94 CO-02 - Utiliser uniquement les portions indispensables des frameworks CSS

Catégorie : - Code - CSS

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

94.1 Description :

Les frameworks CSS prêts à l'emploi (Bootstrap, skeleton, gumby, foundation...) sont d'excellents outils pour réaliser rapidement des sites, car ils répondent à presque tous les besoins les plus courants. Revers de la médaille, on n'utilise généralement qu'une petite portion de ces frameworks. Or, ces frameworks ne s'appuient pas tous sur une architecture modulaire, ce qui contraint l'internaute à télécharger la librairie entière pour n'utiliser qu'un faible pourcentage de ses fonctionnalités.

94.2 Mise en œuvre :

En tant que développeur, dans la mesure du possible, je n'ai pas recours à des frameworks ou je n'en conserve que les portions réellement utilisées (voir <https://getbootstrap.com/customize>).

Si une contrainte m'oblige à utiliser des librairies ou framework pour mon projet, j'opte pour les alternatives légères qui répondent au mieux à mon besoin. Un travail est en cours pour répertorier les librairies utilisées dans les projets de Pôle emploi, et proposer des alternatives plus sobres.

94.3 Tests d'acceptance :

Le nombre de frameworks utilisés est inférieur ou égal à 1.

95 CO-03 - Utiliser des sélecteurs CSS efficaces

Catégorie : - Code - CSS

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 1

95.1 Description :

Un sélecteur est une expression qui indique au navigateur à quelle entité HTML applique la règle CSS correspondante. Toute règle CSS commence par un sélecteur. CSS propose une grande diversité de sélecteurs (sélecteurs de type, de classe, d'ID, d'attributs, etc.), offrant ainsi une précision très fine dans la façon de cibler les éléments.

Afin de limiter l'impact environnemental généré par les sélecteurs, il est recommandé de privilégier les sélecteurs basés sur des ID ou des classes. En effet, ces sélecteurs permettent de filtrer très rapidement les éléments HTML, économisant ainsi des cycles CPU à la machine interprétant les règles et donc les impacts environnementaux.

95.2 Mise en œuvre :

En tant que développeur :

A la place de `treeitem[mailfolder="true"] > treerow > treecell {...}`

J'écris `.treecell-mailfolder {...}`

A la place de `treehead > treerow > treecell {...}`

j'écris `.treecell-header {...}`

95.3 Tests d'acceptance :

Le nombre de sélecteurs CSS qui n'utilisent ni d'id, ni de class et ni d'attribut est égal à 0.

96 CO-04 - Grouper les déclarations CSS similaires

Catégorie : - Code - CSS

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 1

96.1 Description :

Lorsque plusieurs éléments du DOM (Document Object Model) ont des propriétés CSS communes, il est recommandé de les déclarer ensemble dans la même feuille de styles. Cette méthode permet de réduire le poids de la CSS, et donc l'impact environnemental induit par le traitement, le transport et le stockage des données.

96.2 Mise en œuvre :

En tant que développeur, à la place de :

```
h1 {  
background-color: gray; color: navy;  
}  
h2 {  
background-color: gray; color: navy;  
}  
h3 {  
background-color: gray; color: navy;
```



```
}
```

J'écris :

```
h1, h2, h3 {  
background-color: gray; color: navy;  
}
```

Pour s'assurer que cette règle soit bien respectée, il est possible de mettre en place le module stylelint¹ dans le projet (devDependencies) et utiliser la règle shorthand-property-no-redundant-values

96.3 Tests d'acceptance :

Le nombre de déclaration CSS similaires non regroupées est inférieur ou égal à 1.

1. <https://stylelint.io/>

97 CO-05 - Modifier plusieurs propriétés CSS en une seule fois

Catégorie : - Code - CSS

Phase : Build

Responsable : Développeur

Bénéfices : 1

Priorité : 2

97.1 Description :

Le repaint est le changement d'apparence d'un élément du DOM (Document Object Model), tandis que le reflow est le changement/recalcul de la position des éléments dans le DOM. Ces deux opérations sont coûteuses en ressources énergétiques, notamment en cycles CPU : il faut donc éviter de les déclencher. Pour limiter le nombre de repaint/reflow, il est conseillé de ne pas modifier des propriétés une à une. Préférer l'ajout/la suppression de classes CSS, ce qui permet de modifier en une seule fois plusieurs propriétés, tout en ne générant qu'un repaint/reflow.

97.2 Mise en œuvre :

En tant que développeur, je favorise ce type d'écriture :

```
<style>
    .in-error { color: red;
                font-weight: bold;}
</style>
<script>
    $el.bind('error', function () {
```

```
        $.addClass('in-error');});  
$.bind('running'), function () {  
    $.removeClass('in-error');});  
</script>.
```

97.3 Tests d'acceptance :

Le nombre de déclarations CSS pouvant être regroupées en une seule (margin, padding, ...) n'utilisant pas la version d'écriture courte est égal à 0

98 CO-06 - Prévenir le navigateur du déclenchement d'une animation

Catégorie : - Code - CSS

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 1

98.1 Description :

La propriété CSS will-change permet au navigateur d'anticiper les propriétés et les éléments qui vont être manipulés sur la page, et donc d'améliorer les performances des opérations concernées.

Cette propriété peut prendre 4 valeurs : - auto : une optimisation standard va être appliquée par le navigateur, - scroll-position : permet d'indiquer que l'élément va voir sa position de défilement modifiée, le navigateur va donc pouvoir préparer l'affichage d'éléments qui ne figurent pas encore dans le cadre de visualisation - contents : permet de déclarer que le contenu de l'élément sera amené à changer, le navigateur ne va donc pas mettre ce contenu en cache - <custom> : peut contenir n'importe quelle propriété CSS comme, transform ou font-size, sur lesquelles le will-change sera appliqué. Plusieurs valeurs peuvent être déclarées, en les séparant par une virgule (exemple : will-change: transform, opacity;).

L'instruction will-change peut néanmoins causer des problèmes de performance si elle est trop utilisée. Ainsi, elle ne doit pas être appliquée à tous les éléments d'une page et en prévision d'optimisations, mais seulement sur les éléments qui engendrent des problèmes de performance et d'affichage.

98.2 Mise en œuvre :

En tant que développeur, j'utilise l'instruction will-change sur les éléments qui engendrent des problèmes de performance et d'affichage.

98.3 Tests d'acceptance :

La propriété will-change a été utilisée sur les éléments la nécessitant.

99 CO-07 - Concaténer les fichiers Javascript

Catégorie : - Code - Javascript

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 2

99.1 Description :

La combinaison de bibliothèques Javascript permet de réduire le nombre de requêtes HTTP et donc les impacts environnementaux associés (cf fiche ST-13). Cette bonne pratique ne sera plus valable si Pôle emploi adopte le protocole HTTP/2, qui concatène déjà les fichiers CSS.

99.2 Mise en œuvre :

En tant que développeur, j'utilise le plugin webpack-merge-and-include-globally de l'outil Webpack pour fusionner les fichiers Javascript.

99.3 Tests d'acceptance :

Il y a au maximum 10 fichiers Javascript

100 CO-08 - Utiliser uniquement les portions indispensables des librairies JavaScript

Catégorie : - Code - Javascript

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

100.1 Description :

Les bibliothèques JavaScript telles que jQuery sont d'excellents outils pour réaliser rapidement des sites, car ils répondent à presque tous les besoins les plus courants. Revers de la médaille, on n'en utilise généralement qu'une petite portion ; or ces bibliothèques ne s'appuient pas tous sur une architecture modulaire, ce qui contraint l'internaute à télécharger toute la librairie pour n'utiliser qu'un faible pourcentage de ses fonctionnalités.

100.2 Mise en œuvre :

En tant que développeur, dans la mesure du possible, je n'ai pas recours à des bibliothèques (ex : je peux me baser sur le code de <https://youmightnotneedjquery.com> à la place d'utiliser jquery) ou je n'en conserve que les portions réellement utilisées.

Si une contrainte m'oblige à utiliser des librairies pour mon projet, j'opte pour les alternatives légères qui répondent au mieux à mon besoin. Un travail est en cours pour répertorier les librairies utilisées dans les projets de Pôle emploi, et proposer des alternatives plus sobres.

100.3 Tests d'acceptance :

Le nombre de bibliothèques dont des portions non indispensables sont utilisées est inférieur ou égal à 1.

101 CO-09 - Utiliser un framework javascript adapté et efficient

Catégorie : - Code - Javascript

Phase : Build

Responsable : Equipe produit

Bénéfices : 2

Priorité : 3

101.1 Description :

En 2021, le poids médian d'un site web représente 1 923 KB. Les images et le code JavaScript sont les éléments qui contribuent en majeure partie à ce poids. La plupart des projets qui utilisent de grandes quantités de JavaScript utilisent une bibliothèque ou un framework. Il existe une multitude de frameworks Javascript, chacun ayant ses caractéristiques propres. Chaque framework ne répond pas forcément aux mêmes critères et besoins. Afin de maximiser l'efficacité d'un framework vis-à-vis d'un besoin, et donc limiter ses impacts environnementaux, il faut choisir un framework qui est conçu pour répondre à ce besoin. Parmi les framework adaptés au besoin et maîtrisés par l'équipe, il conviendra d'utiliser le plus efficient. Un classement des frameworks les plus populaires en fonction de leur performance environnementale est disponible ici. Parmi les frameworks étudiés, Vanilla JS¹ semble être le plus performant, suivi de près par Solid, Lit et Svelte. Angular et React sont les moins performants des frameworks étudiés.

101.2 Mise en œuvre :

En tant qu'équipe produit, lors du cadrage d'architecture, je détermine le framework à utiliser en prenant en compte sa performance, le contexte du produit et les compétences de l'équipe.

1. <http://vanilla-js.com/>

101.3 Tests d'acceptance :

- Le critère de performance a été pris en compte dans le choix du framework lors du cadrage d'architecture
- L'adéquation au besoin a été pris en compte dans le choix du framework lors du cadrage d'architecture.

102 CO-10 - Ne pas copier des tableaux via des boucles for, while, do..while et foreach

Catégorie : - Code - Java

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

102.1 Description :

Utiliser une boucle pour copier un tableau ou une partie d'un tableau rajoute des lignes de code qui auraient pu être évitées en utilisant des fonctions natives, naturellement optimisées par le langage. Ce code supplémentaire induit une consommation non nécessaire de ressources techniques, qui se traduit par une augmentation de l'empreinte environnementale du service numérique.

102.2 Mise en œuvre :

En tant que développeur, j'utilise les méthodes natives pour copier les tableaux : - Si je veux copier un tableau dans un autre tableau, j'utilise la méthode `Arrays.copyOf` - Si je veux copier une partie d'un tableau dans un autre tableau, j'utilise la méthode `System.arraycopy` - Si je veux alimenter une liste à partir d'un tableau existant, j'utilise la méthode `Arrays.asList`. Des étapes supplémentaires sont à prévoir si une liste à taille variable est désirée

102.3 Tests d'acceptance :

Pour copier un tableau, l'une des méthodes suivantes a été utilisée : `Arrays.copyOf`, `System.arraycopy` ou `Arrays.asList`

103 CO-11 - Éviter d'utiliser SELECT * FROM

Catégorie : - Code - Java

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

103.1 Description :

Bien souvent, seuls certains champs du tableau sont réellement nécessaires pour un cas d'usage. Cependant, une requête du type `SELECT *` récupère l'ensemble des données d'un tableau. Ainsi, ce type de requêtes collecte souvent plus de champs de la base de données que ce dont le service numérique a réellement besoin pour fonctionner. Cela entraîne une augmentation du volume de données à déplacer du serveur de base de données vers le client, ce qui ralentit l'accès et augmente la charge des machines, tout en prenant plus de temps pour traverser le réseau.

Dès lors, si les champs utiles sont connus, il est vivement recommandé de nommer les champs dans la requête et de détailler au maximum la requête pour récupérer un minimum d'informations.

103.2 Mise en œuvre :

En tant que développeur, à la place de :

```
SELECT * FROM clients;
```

J'écris :

```
SELECT raison_social, adresse, code_postal, telephone FROM clients;
```

Les clauses telles que `WHERE`, `LIMIT` ou encore `LIKE` par exemple peuvent aussi permettre d'affiner une requête.

103.3 Tests d'acceptance :

Le nombre d'occurrence de « `SELECT * FROM` » dans le code est égal à 0 sauf cas exceptionnel (par exemple si une table contient peu de lignes et de colonnes).

104 CO-12 - Éviter d'utiliser « EXECUTE IMMEDIATE »

Catégorie : - Code - Transverse

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 2

104.1 Description :

EXECUTE IMMEDIATE prépare et exécute immédiatement un ordre SQL spécifié dynamiquement, sans récupérer les enregistrements du résultat. Les résultats doivent donc être recalculés à chaque fois qu'ils sont utilisés. Cela induit des consommations de ressources et donc des impacts environnementaux supplémentaires.

104.2 Mise en œuvre :

En tant que développeur, je remplace les instructions SQL dynamiques par des instructions statiques.

104.3 Tests d'acceptance :

La fonction SQL « EXECUTE IMMEDIATE » n'est pas utilisée.

105 CO-13 - Eviter de faire des requêtes SQL à l'intérieur d'une boucle

Catégorie : - Code - Transverse

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 2

105.1 Description :

Les requêtes SQL à l'intérieur d'une boucle posent des problèmes importants de performance, et ce d'autant plus si le(s) serveur(s) SQL n'est (ne sont) pas sur la machine locale. En effet, ces serveurs sont optimisés pour traiter plusieurs sélections, insertions ou modifications dans une seule requête ou une seule transaction. Ainsi, envoyer un grand nombre de petites requêtes constitue le meilleur moyen de consommer inutilement des cycles CPU, de la mémoire vive et de la bande passante, et ainsi induire des impacts environnementaux non négligeables. Cela peut par ailleurs saturer un serveur.

105.2 Mise en œuvre :

En tant que développeur, à la place d'écrire :

```
for (var user in userList) {  
    const query = "INSERT INTO users (first_name,last_name)  
VALUES("+user.first_name+", "+user.last_name+")";  
    const res = db.exec(query);  
}
```

J'écris :

```
var sql = "INSERT INTO Test (first_name, last_name) VALUES ?";
var values = [
    ['demian', 'smith'],
    ['john', 'doe'],
    ['mark', 'lee'],
    ['pete', 'buler']
];
conn.query(sql, [values], function(err) {
    if (err) throw err;
    conn.end();
});
```

105.3 Tests d'acceptance :

Le nombre de requêtes SQL à l'intérieur d'une boucle est égal à 0.

106 CO-14 - Ne pas réécrire les getter et setter natifs

Catégorie : - Code - Transverse

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 2

106.1 Description :

La plupart des langages objet proposent des getter et des setter standards que le développeur n'a pas besoin d'écrire. Réécrire à sa manière les getter et setter standards rajoute des lignes de code qui auraient pu facilement être évitées. Cela allonge les temps de compilation et d'exécution de ces méthodes, qui sont généralement bien mieux optimisées par le langage que par le développeur.

106.2 Mise en œuvre :

En tant que développeur, j'utilise les méthodes natives get et set, et j'évite à tout prix de les réécrire. Des exemples sont présentés ci dessous :

```
class Car {
    constructor(name) {this.name = name;}
    get name() {return this._name;}
    set name(newName) {this._name = newName;}
    /* non native */
    getName() {return this._name;}
    setName() {return this.name;}}
```

```
let my_first_car = new Car("Ferrari");

/* getter */
console.log('My car\'s name with non native getter : ', my_car.getName()) // no
console.log('My car\'s name with native getter : ', my_car.name) // correct
console.log('My car\'s name : ', my_car._name) // correct

/* setter */
my_car.setName("Mercedes"); // non natif
console.log('My car\'s name setted with non native getter : ', my_car.name) //

my_car.name = "Renault";
console.log('My car\'s name setted with native getter : ', my_car.name) // nati
```

106.3 Tests d'acceptance :

Le nombre de getter et setter natifs réécrit est égal à 0.

107 CO-15 - Privilégier les variables locales

Catégorie : - Code - Transverse

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 2

107.1 Description :

En informatique, les variables sont des symboles qui associent un nom à une valeur. Une variable globale est définie en dehors de toute fonction et est par conséquent accessible et partagée par toutes les fonctions. A contrario, une variable locale est définie à l'intérieur d'un bloc (en général d'une fonction) et a une durée de vie limitée à l'exécution du bloc (de fonction), elle est visible uniquement dans ce bloc, inconnue en dehors.

En JavaScript, lorsqu'on fait appel à une variable globale, le moteur d'interprétation doit vérifier : - qu'elle existe dans le scope actuel, dans celui du dessus, etc. ; - que la variable dispose d'une valeur, etc.

Pour éviter toutes ces vérifications, il est souvent envisageable de passer les variables utiles en arguments des routines, les rendant locales. Ce procédé permet ainsi d'économiser du temps de calcul (cycles CPU), et donc les impacts environnementaux qui en découlent.

107.2 Mise en œuvre :

En tant que développeur, à la place de :

```
var global_variable = 0;  
function foo(){
```

```
    global_variable = 0;
    for (var i=0; i<items.length; i++) {
        global_variable += items[i];
    }
}
```

J'écris :

```
function foo(){
    var local_variable = 0;
    for (var i=0, li=items.length; i<li; i++) {
        local_variable += items[i];
    }
    return local_variable;}
}
```

107.3 Tests d'acceptance :

Le nombre de variables globales utilisées dans le code est égal à 0.

108 CO-16 - Supprimer le code inutilisé

Catégorie : - Code - Méthode et outils

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

108.1 Description :

L'exécution d'un code induit un transfert et un traitement de données, qui engendrent à leur tour des impacts environnementaux. Avoir du code inutilisé induit alors des impacts environnementaux inutiles.

Par ailleurs, réduire le code permet de limiter les bugs et les problèmes de performance lors de la modification du code.

Dès lors, l'intérêt de garder du code inutilisé est limité, d'autant plus que la probabilité qu'il soit réutilisé un jour est très faible.

108.2 Mise en œuvre :

En tant que développeur, sur chrome, le panneau DevTools Couverture m'aide à localiser le code CSS et JavaScript inutilisé (ligne par ligne). D'autres outils me permettent d'identifier et de supprimer spécifiquement le CSS inutilisé, en recopiant les codes HTML et CSS de ma page, par exemple : UnCSS¹, UnusedCSS, PurifyCCC, etc.

1. <https://uncss-online.com/>

108.3 Tests d'acceptance :

- Le taux de CSS inutilisé est inférieur à 70%
- Le taux de JS inutilisé est inférieur à 70%

109 CO-17 - Minifier le code

Catégorie : - Code - Méthode et outils

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

109.1 Description :

Minifier un code consiste à réduire sa taille au maximum tout en conservant l'intégralité des informations contenus dans ce dernier. Concrètement, la minification correspond à : - Supprimer les commentaires - Supprimer les caractères "vides" (espaces, tabulations, retour à la ligne) - Raccourcir les noms de variables et fonctions

Ainsi, en réduisant le poids des fichiers de code, la minification contribue à la réduction de l'empreinte environnementale des services numériques.

109.2 Mise en œuvre :

En tant que développeur, plusieurs outils me permettent de minifier un fichier CSS ou JavaScript en ligne ou en ligne de commande tels que : - Minifier - uglify-js - Minify - JSCompress.com

109.3 Tests d'acceptance :

100% des fichiers sont minifiés.

110 CO-18 - Ecrire un code propre

Catégorie : - Code - Méthode et outils

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

110.1 Description :

Un code propre est un code simple et facilement compréhensible par tout développeur. Ainsi, un code propre est facilement maintenable et dure dans le temps.

110.2 Mise en œuvre :

En tant que développeur, j'applique les principales caractéristiques d'un code propre : - Mon code doit être simple :

Concis et efficace

Intuitif et facile à comprendre

Aucune répétition inutile

— Mon code doit être structuré :

Indentations

Découpage logique du programme

Les fonctions, classes, variables doivent porter des noms explicites

Respect des conventions d'écriture

— Mon code doit être commenté

110.3 Tests d'acceptance :

- Le code est simple
- Le code est structuré
- Le code est commenté

111 CO-19- Utiliser le validateur du W3C

Catégorie : - Code - Méthode et outils

Phase : Build

Responsable : Développeur

Bénéfices : 1

Priorité : 3

111.1 Description :

Vérifier que le code HTML des pages est bien formé. Dans le cas contraire, le navigateur corrigera dynamiquement un certain nombre d'éléments pour afficher au mieux les pages posant problème. Ces corrections dynamiques consomment inutilement des ressources à chaque chargement des pages concernées.

111.2 Mise en œuvre :

En tant que développeur, j'utilise le validateur du W3C ¹(World Wide Web Consortium) pour vérifier que les pages sont bien valides et que le code HTML est correctement formé.

111.3 Tests d'acceptance :

- Le validateur du W3C a été utilisé pour vérifier le code HTML
- Le nombre d'erreurs critiques est égal à 0

1. <https://validator.w3.org/>

112 CO-20- Valider le code avec un linter

Catégorie : - Code - Méthode et outils

Phase : Build

Responsable : Développeur

Bénéfices : 1

Priorité : 2

112.1 Description :

Un analyseur de la conformité de la syntaxe du code vérifie que la syntaxe du code est correcte. Le code obtenu respecte ainsi des contraintes syntaxiques qui permettent aux interpréteurs d'exécuter le code plus facilement et donc plus rapidement. Le processeur est donc sollicité moins longtemps.

112.2 Mise en œuvre :

En tant que développeur, des outils de validation de code sont à ma disposition, par exemple : - ESLint ou JSLint pour le code JavaScript - Stylelint pour les feuilles de styles

112.3 Tests d'acceptance :

- Un linter a été utilisé
- Le nombre de lignes de code source non validées par un linter est égal à 0

113 CO-21- Utiliser des outils d'analyse de code statique

Catégorie : - Code - Méthode et outils

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

113.1 Description :

Les outils d'analyse de code statique évaluent, compilent et vérifient en phase de test la qualité du code (vulnérabilités, erreurs, normes, etc.). Cela permet en particulier de repérer des erreurs de programmation ou de conception et témoigne de la maintenabilité du code.

113.2 Mise en œuvre :

En tant que développeur, j'utilise l'outil SonarQube pour vérifier la qualité de mon code.

113.3 Tests d'acceptance :

SonarQube a été utilisé pour les tests.

114 CO-22- Utiliser des outils d'analyse de code dynamique

Catégorie : - Code - Méthode et outils

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

114.1 Description :

L'analyse de code dynamique est une forme d'analyse de programme qui nécessite leur exécution. Elle permet d'étudier le comportement d'un programme informatique et les effets de son exécution sur son environnement. Le profiling ou le débogage sont des formes d'analyse de code dynamique.

Le débogage est un processus de diagnostic, de localisation et d'élimination des erreurs des programmes informatiques.

Le profilage est réalisé en instrumentant le code source du programme ou sa forme exécutable binaire à l'aide d'un outil appelé profileur. Il permet de mesurer, par exemple : - l'utilisation mémoire - l'utilisation processeur - la fréquence et la durée des appels de fonction

Les informations de profilage servent à identifier les parties du code qu'il faut optimiser. Exemple d'outil de profilage : Java visual VM (backend).

114.2 Mise en œuvre :

En tant que développeur, j'utilise des outils d'analyse de code dynamique pendant la phase de développement.

114.3 Tests d'acceptance :

Le développeur a utilisé un outil d'analyse de code dynamique pendant la phase de développement.

115 CO-23- Vérifier les erreurs de code dans la console

Catégorie : - Code - Méthode et outils

Phase : Build

Responsable : Développeur

Bénéfices : 1

Priorité : 3

115.1 Description :

Le code peut contenir des erreurs de syntaxe ou de logique. Il peut parfois être difficile d'identifier et de comprendre ces erreurs. Heureusement, la plupart des navigateurs disposent d'un outil pour gérer les erreurs de code Javascript, CSS et HTML.

115.2 Mise en œuvre :

En tant que développeur, dans l'outil d'aide aux développeurs (F12), je me rends sur l'onglet « Console » pour afficher les erreurs présentes sur la page web.

115.3 Tests d'acceptance :

- Le nombre d'erreurs est vérifié régulièrement dans la console
- Le nombre d'erreurs dans la console est égal à 0.

116 CO-24- Réutiliser le code développé en interne

Catégorie : - Code - Méthode et outils

Phase : Build

Responsable : Equipe produit

Bénéfices : 3

Priorité : Optionnel

116.1 Description :

L'innersource consiste à utiliser les pratiques de développement open source et à mettre en place une culture type open source au sein des entreprises. Concrètement, l'innersource a pour objet de partager au sein d'une entreprise le code de différents projets ou encore de la documentation afin que chacun puisse y apporter sa contribution. Ainsi, à l'inverse de l'Open source, l'accès aux données n'est pas public mais se veut strictement réservé à l'entreprise.

Dans un système d'innersource, un code n'est développé qu'une seule fois, ce qui induit : - Un code de meilleure qualité, qui favorise la résilience du service numérique - Une recherche et une réutilisation du code facilitée à grande échelle, qui évite le gaspillage de ressources et la duplication, et limite donc les impacts environnementaux

D'autres avantages peuvent être relevés comme par exemple un gain financier ou des cycles de développement accélérés.

Chez Pôle emploi, l'innersource est mis en place au travers de : - Un climat de confiance et une communication claire - Canaux de communication asynchrone : Teams (en cours d'installation) et Mattermost ¹ - Des règles de contribution définies clairement (cf. le fichier contributing.md) - Des outils - Repository privé : Gitlab ² - Base de connaissance : Confluence ³ - Système de bug tracking et de gestion de projets

1. <https://mattermost.pe-qvr.net/>

2. <https://git-scm.pole-emploi.intra/>

3. <https://confluence.pole-emploi.net/>

: RTC puis Jira - Portail des projets inner source ⁴ qui recense les projets innersource - Une chaîne d'intégration continue mature - Une bonne connaissance des compétences des collaborateurs

116.2 Mise en œuvre :

En tant qu'équipe produit, je m'assure que j'ai la maturité suffisante pour passer mon projet en inner source en me référant à la checklist innersource ⁵. Ensuite, un template détaillant les étapes à mettre en œuvre est mis à ma disposition ici ⁶.

116.3 Tests d'acceptance :

- L'équipe produit a mesuré sa maturité innersource
- L'équipe produit a suivi les étapes détaillées dans le template de mise en œuvre

4. <http://projets-innersource.media.k8s.pole-emploi.intra/#!/&display=card>

5. https://git-scm.pole-emploi.intra/innersource/innersource.git-scm.pole-emploi.intra/blob/master/Innersource_CheckList.md

6. <https://git-scm.pole-emploi.intra/innersource/template-projet-innersource>

117 CO-25- Utiliser l'open source de manière responsable

Catégorie : - Code - Méthode et outils

Phase : Build

Responsable : Equipe produit

Bénéfices : 3

Priorité : Optionnel

117.1 Description :

La désignation « open source », ou « code source ouvert », s'applique aux services numériques dont la licence offre les possibilités suivantes : libre redistribution, mise à disposition du code source au grand public et création de travaux dérivés. Une grande communauté de développeurs assure le support et l'amélioration continue des produits open source, grâce aux contributions collectives.

Ainsi, un développeur peut de manière autonome traiter un besoin spécifique sans recours à l'éditeur. Le risque d'obsolescence par manque de support technique, l'absence volontaire de mises à jour, ou par volonté de pousser à la consommation est donc résolu. Le coût environnemental lié au développement d'un nouveau service numérique est donc évité.

Tout comme une part croissante des entreprises, Pôle emploi est actuellement en train de structurer sa démarche de contribution à l'externe.

Il faut néanmoins faire preuve de précaution par rapport aux outils open source du marché. En effet, face à l'engouement des entreprises pour l'open source, une quantité importante de services numériques en open source a été rendue disponible. Tous ces services ne sont pas forcément optimisés et embarquent possiblement des fonctionnalités qui ne sont pas utiles pour tous les cas d'usage. Ces services numériques, plus lourds, consomment davantage des ressources et induisent donc des impacts environnementaux plus importants.

117.2 Mise en œuvre :

En tant qu'équipe produit, lorsque je fais appel à j'ai un service numérique externe, je privilégie l'utilisation d'un service open source, en ayant vérifié au préalable qu'il ne contient pas trop de fonctionnalités inutiles par rapport à mon cas d'usage. J'évalue la pertinence d'ouvrir un service numérique à l'extérieur avant de le faire en me posant plusieurs questions : - Est-ce que le service numérique est utile ? - A qui sera-t-il utile ? - Pourquoi ?

Si les réponses à ces questions prouvent la pertinence de l'ouverture du service numérique, alors je peux le faire.

117.3 Tests d'acceptance :

- Une solution open source a été privilégiée à condition qu'elle soit frugale
- En amont de l'ouverture d'un service numérique, la pertinence de cette ouverture a été évaluée

118 CO-26- Utiliser la CI/CD de manière responsable

Catégorie : - Architecture - Principes

Phase : Build

Responsable : Devops

Bénéfices : 2

Priorité : 3

118.1 Description :

L'adoption du Devops peut entraîner une consommation de ressources largement supérieure au besoin si elle n'est pas encadrée. Optimiser l'utilisation des ressources informatiques est nécessaire afin de limiter leur consommation.

118.2 Mise en œuvre :

En tant que devops, plusieurs bonnes pratiques me permettent d'utiliser la CI/CD de manière responsable :

- Optimiser la taille des images Docker (fichier utilisé pour exécuter du code dans un conteneur Docker) utilisées afin de limiter les transits réseaux, le stockage et la latence lors du déploiement d'une nouvelle instance d'image. Plusieurs techniques ¹ permettent de réduire la taille des images Docker. Cette bonne pratique est préconisée dans TMJ, un mode opératoire ² a même été rédigé à ce sujet
- Mettre en cache les fichiers et dossiers tout au long de la CI/CD afin d'éviter de retélécharger les fichiers à chaque build. Ce système de cache existe chez Pôle emploi, mais est à optimiser. Un

1. <https://enix.io/fr/blog/cherie-j-ai-retréci-docker-part1/>

2. http://tmj.pe.intra/site/livrables_tmj/conteneurisation/creer-son-image-docker/README.html

projet est en cours pour augmenter la probabilité qu'un fichier s'exécute sur le même worker où il s'était exécuté auparavant (à titre d'illustration, il y a 96 workers dans Concourse)

- Surveiller la durée des jobs, leur nombre, et le trafic réseau et les événements dans Grafana et surveiller le volume des artefacts dans Artifactory afin de détecter et remédier au plus tôt aux défauts fonctionnels, bugs et surconsommations d'énergie, et ainsi améliorer l'efficacité des workflows
- Privilégier les forges mutualisées (solutions permettant de centraliser les informations d'un projet et intégrant un processus d'intégration continue permettant de compiler, tester et déployer un ou plusieurs projets) -par exemple, l'outil Concourse est utilisé chez Pôle emploi- afin de mutualiser et donc d'augmenter le taux d'utilisation des serveurs. D'autres outils sont utilisés : GCLP, Proven, ARA, Gitlab CI et Jenkins
- Assurer une cohérence globale à l'ensemble des outils de manière à optimiser l'espace disque. Un chantier de nettoyage des données dans les différents outils a été identifié par Pôle emploi et sera prochainement mis en œuvre

118.3 Tests d'acceptance :

- La taille des images Docker est optimisée
- Un mécanisme de mise en cache optimisé est intégré à la CI/CD
- Les métriques de pilotage sont surveillées quotidiennement
- Les données sont supprimées dès qu'elles deviennent obsolètes

119 AR-01- Utiliser la CI/CD de manière responsable

Catégorie : - Architecture - Données

Phase : Cycle de vie entier

Responsable : Data steward, administrateur des données

Bénéfices : 1

Priorité : 3

119.1 Description :

La classification des données est une action de catégorisation des données, en fonction du type et du contenu du fichier. Elle permet notamment de définir comment une donnée va être exploitée (durée de vie, duplication, etc.) en fonction de la catégorie à laquelle elle appartient.

La cartographie des données est une plateforme qui recense les données, leur origine et leur usage. Elle donne une vision globale et en temps réel des données d'une entreprise, permettant ainsi de faciliter leur gestion.

La cartographie et la classification des données sont un support nécessaire pour assurer la protection des données personnelles et, plus généralement, elles permettent aux entreprises de maîtriser leur patrimoine de données.

Chez Pôle emploi, des travaux côté Data Gouv sont en cours pour connaître les datas et leur positionnement dans le SI. La solution choisie par le volet Data Gouv pour documenter à terme les données est Zeenea. Le catalogue de données permet de documenter le patrimoine de données dans la cible SI de Pôle emploi. Il est disponible en production, et les données y seront chargées progressivement jusqu'à fin 2023.

Aujourd'hui, la living doc (documentation vivante), capacité déployée à l'ensemble des équipes fabricantes (et notamment Zenit), permet de capitaliser de la documentation sur les données cœur de métier.

119.2 Mise en œuvre :

A définir, projets en cours.

119.3 Tests d'acceptance :

- Il n'existe pas de donnée non cartographiée
- Il n'existe pas de donnée non classifiée

120 AR-02- Vérifier l'existence des données avant de créer une structure de données ad hoc

Catégorie : - Architecture - Données

Phase : Cycle de vie entier

Responsable : Maître d'ouvrage ou chef de projet

Bénéfices : 1

Priorité : 2

120.1 Description :

Lors de l'étape de conception d'un projet numérique, une structure de données ad hoc est souvent créée sans s'assurer que les données soient déjà disponibles par ailleurs. Cela induit une redondance de données et donc des impacts environnementaux supplémentaires qui auraient pu être évités en se questionnant sur la disponibilité ou non des données dans le panorama actuel.

120.2 Mise en œuvre :

Si les données existent déjà dans le panorama actuel, il est préférable de les réutiliser. Avant de créer une structure de données de toutes pièces, il y a plusieurs façons de vérifier leur disponibilité :

- L'équipe data doit consulter le catalogue de données. A la cible (2023), il offrira l'avantage de proposer une vue exhaustive des données du patrimoine de l'Etablissement, quelles que soient les sources. Il s'adosse à l'ensemble des dictionnaires de données (Lac, SISP...) pour apporter d'avantage d'information; ainsi qu'une approche sémantique, reflet des concepts métier manipulant ces données. Ainsi, la recherche de l'existence d'une donnée peut se faire simplement via le moteur de recherche qui intègre les objets tant applicatifs que fonctionnels et métier. Enfin, il permet d'identifier les points de contact, responsables des données.
- Consulter les dictionnaires de données, d'une approche plus technique que le catalogue.

- Questionner les architectes fonctionnels et applicatifs pour une approche organisationnelle des objets applicatifs.

120.3 Tests d'acceptance :

La disponibilité des données a été vérifiée dans le dictionnaire de données, le catalogue de données ou auprès d'un architecte.

121 AR-03- Réutiliser les datasets

Catégorie : - Architecture - Données

Phase : Cycle de vie entier

Responsable : Maître d'ouvrage ou chef de projet

Bénéfices : 1

Priorité : 3

121.1 Description :

Un dataset (ou jeu de données) est un « container » qui stocke des données, du code ainsi que la documentation associée et qui est décrit par des métadonnées.

Chez Pôle emploi, les données nécessaires aux cas d'usages sont dupliquées dans des datasets. Or, différents cas d'usages peuvent nécessiter des données communes. Au lieu de recréer des datasets pour chaque cas d'usage et ainsi dupliquer les données, il faut réutiliser les datasets existants. La réutilisation de datasets peut concerner les datasets d'entrée d'un cas d'usage (CU) ou des datasets « enrichis » en cours de CU :

— Réutilisation de datasets d'entrée :

Chaque cas d'usage est documenté par une fiche qui liste l'ensemble des tables nécessaires à sa réalisation. Ces CU sont passés en revue lors du Comité de Suivi des Cas d'Usage (CSCU – cf. colibri n° 351 264 ¹), qui peut détecter une adhérence entre des projets et ainsi les mettre en relation pour un partage des tables consommées en entrée. Par ailleurs, les cas d'usages accompagnés par l'ODS Datalab ² sur la plateforme Dataiku sont sensibilisés à la problématique de réutilisation et de stockage des données, lors des démos et formations organisées par l'équipe support aux dev. En effet, afin d'éviter de nombreuses duplications et stockages inutiles, les tables du Datalake et de SISP ne sont ni importées ni copiées dans l'outil, mais sont consultables en lecture. Seul le résultat des requêtes adressées est transporté et écrit dans le lac.

1. <https://colibri.pole-emploi.net/nuxeo/site/getDocumentPermalink?numId=351264>

2. <mailto:datalab.00326@pole-emploi.fr>

— Réutilisation de datasets « enrichis » :

Dans un groupe de cas d'usages ayant un périmètre ou une finalité proche, l'un des CU peut avoir à utiliser un ou plusieurs dataset(s) généré(s) par un autre CU. Cette identification se fait souvent par connaissance de ces projets. Ces datasets peuvent aussi être facilement identifiables pour les CU accompagnés par le Datalab. En effet lors des formations, la fonction « Catalog » de Dataiku est mise en avant, permettant ainsi de visualiser l'ensemble des datasets des cas d'usage d'un groupe. Le partage de ces datasets est ensuite encadré afin de protéger et de sécuriser l'accès aux données. Ainsi, une demande doit être faite à l'équipe support aux dev du Datalab qui, une fois la légitimité vérifiée lors d'un CSCU, configure les accès à ces datasets. Au-delà du seul aspect de partage d'un dataset unique, le partage de datasets enrichis induit une économie liée aux autres datasets intermédiaires et aux traitements nécessaires à son obtention.

121.2 Mise en œuvre :

En tant que MOA ou chef de projet, selon les cas : - Je documente les tables nécessaires à la réalisation d'un cas d'usage et je partage ces tables avec les projets ayant des adhérences identifiées en CSCU - Je vérifie via le Catalog de l'outil Dataiku les datasets existants des CU de mon groupe

121.3 Tests d'acceptance :

- Une fiche cas d'usage avec les tables consommées a été documentée pour revue CSCU
- Les datasets produits ont été partagés avec les cas d'usage d'un même groupe
- Les datasets existants des CU du groupe ont été vérifiés via le Catalog de l'outil Dataiku

122 AR-04- Définir une politique de cycle de vie des données et l'appliquer

Catégorie : - Architecture - Données

Phase : Cycle de vie entier

Responsable : Architecte logiciel / Développeur / Métier

Bénéfices : 1

Priorité : 3

122.1 Description :

Le cycle de vie d'une donnée correspond à l'enchaînement des étapes par lesquelles va passer une donnée tout au long de son existence (de sa collecte / création à sa suppression).

La réglementation RGPD actuelle impose de maîtriser la gestion des données pour certaines catégories (données personnelles encadrées par le RGPD). La gestion du cycle de vie des autres catégories de données n'est pas forcément maîtrisée par les entreprises. Ces données, sans délais de péremption, sont alors accumulées, ce qui contribue donc à un accroissement du volume de données, des volumes de sauvegardes, de ressources consommées pour accéder aux données.

Les équipes doivent donc impérativement définir une politique de cycle de vie des données et l'appliquer. Cela implique notamment de définir une durée de conservation pour chaque catégorie de données. Des outils permettent d'automatiser la suppression des données.

Dans le Lac de données de Pôle emploi, la gestion du cycle de vie des données est réalisée de la façon suivante : - Le métier définit des règles de conservation des données utilisées pour les cas d'usages - Les règles de conservation des données dans le Lac sont les mêmes que celles des cas d'usages métiers associés - Des outils ont été mis en place afin de supprimer automatiquement les données une fois leur date de conservation expirée - => vrai aussi pour le cœur métier

Cependant, à date, le métier ne définit pas toujours les règles de conservation des données. Dans le SISF, en revanche, aucune politique de cycle de vie des données n'a été définie.

122.2 Mise en œuvre :

En tant que collaborateur côté métier, je définis des règles de conservation pour les données que j'ajoute/j'utilise dans le Lac de données.

122.3 Tests d'acceptance :

Des règles de gestion du cycle de vie sont définies pour chaque catégorie de données

123 AR-05- Optimiser la structure du modèle de données

Catégorie : - Architecture - Données

Phase : Cycle de vie entier

Responsable : Architecte logiciel / Développeur

Bénéfices : 1

Priorité : 2

123.1 Description :

Un modèle de données décrit la manière dont sont représentées les données dans une base de données. La modélisation de données permet de comprendre les données et de faire les bons choix technologiques de stockage et de gestion de ces données (un système de gestion de base de données peut prendre en charge un ou plusieurs modèles de base de données).

Un mauvais modèle de données complexifie fortement l'accès à la donnée, et de ce fait induit des problèmes de performances et affecte l'évolutivité du service numérique. Cela va à l'encontre des principes d'écoconception.

123.2 Mise en œuvre :

En tant qu'architecte logiciel / développeur, l'équipe BDD m'accompagne dans la structuration du modèle de données.

Comment la contacter ?

1. Via SNOW (à privilégier) : l'équipe a un catalogue de service dans SNOW

2. Via le Teams DSI – Equipe Produit BDD ¹
3. Via le Mattermost Communauté DBA ²

123.3 Tests d'acceptance :

Des règles de gestion du cycle de vie sont définies pour chaque catégorie de données

1. <https://teams.microsoft.com/l/team/19%3af24e47a14cad45bd90deef1a1a5b50a4%40thread.tacv2/conversations?groupId=5c67fa92-2a30-47a3-b282-e6af836dd86b&tenantId=55a8600f-4ee6-4bb5-8f14-53589536b6df>

2. <https://mattermost.pe-qvr.net/communaute-dba/>

124 AR-06- Optimiser les requêtes et les accès aux données au strict nécessaire

Catégorie : - Architecture - Données

Phase : Cycle de vie entier

Responsable : Architecte logiciel / Développeur / Ops Produit /

Bénéfices : 1

Priorité : 2

124.1 Description :

Les mauvaises performances d'une base de données peuvent par exemple être causées par : - Des requêtes mal rédigées (problème le plus récurrent) - Une mauvaise structure de données - Un problème d'indexation

Or, de mauvaises performances induisent des surconsommations d'énergie et accélèrent l'obsolescence du matériel.

124.2 Mise en œuvre :

En tant qu'architecte logiciel / développeur, je choisis une structure de modèle de données appropriée (cf fiche pratique dédiée).

En tant qu'architecte logiciel / développeur, j'identifie les requêtes ayant un temps d'exécution important. Pour ces requêtes :

- Je combine les requêtes pour lesquelles des données redondantes sont récupérées
- Je permets le chargement progressif (lazy-load) des composants des requêtes

L'optimisation manuelle des requêtes peut s'avérer difficile et fastidieuse, c'est pourquoi il peut s'avérer intéressant d'utiliser un optimiseur de requêtes ou d'externaliser les efforts d'optimisation.

En tant qu'architecte logiciel / développeur, je mène une opération de défragmentation de ma base de données (opération visant à réorganiser le stockage des données et permettant ainsi un accès plus rapide aux données) lorsque mes outils m'en indiquent la nécessité.

En tant qu'Ops produit, je m'assure que les serveurs hébergeant les bases de données disposent à tout moment de suffisamment de ressources en vérifiant la charge CPU, la mémoire et l'espace disque des serveurs associés en cas de problèmes de performance.

124.3 Tests d'acceptance :

Rien ne permet à date de valider la bonne forme d'une requête.

Pour la partie lazy-loading, se référer à la fiche lazy-loading des images (fiche du référentiel numéro UI-01)

125 AR-07- Utiliser un format de données adapté (1/2)

Catégorie : - Architecture - Réseau

Phase : Build

Responsable : Data scientist / Ops Produit /

Bénéfices : 2

Priorité : 3

125.1 Description :

Il existe différents types de formats de fichiers pour échanger de la donnée. Chaque format a ses propres avantages et inconvénients selon les cas d'utilisation et existe pour servir un ou plusieurs objectifs. Voici une liste non exhaustive des formats de fichiers couramment utilisés et de leurs cas d'usages :

Formats	Description	Cas d'usages (non exhaustifs)
XML (eXten-sible Markup Langage)	Format utilisé pour l'échange de données et le stockage structuré d'informations	À moins que la source de données ne l'impose, il faut éviter d'utiliser l'XML en raison de sa grande complexité.
CSV (Comma Separated Values)	Fichier texte qui permet d'échanger des données sous forme de table	- Petits datasets- L'exploration/importation de données- Fichiers de tests

Formats	Description	Cas d'usages (non exhaustifs)
JSON (JavaScript Object Notation)	- Représentation sans schéma, en texte clair, de données structurées basées sur des paires nom/valeur et des listes ordonnées.- Basé sur un sous-ensemble de JS	- Petits datasets- Chargements asynchrone- APIs- Bases de données NoSQL (ex: MongoDB, CouchDB, Riak, etc.)
AVRO	Format de sérialisation de données orienté lignes dans l'écosystème Hadoop	- Big data- Traitement en streaming- Schémas de données qui changent au fil du temps- Utilisable dans un contexte RPC (Remote Procedure Call)- Largement utilisé dans de nombreuses applications (Kafka, Spark, etc.)
Protobuf (Protocol Buffers)	Format de sérialisation de données orienté lignes développé par Google. Il stocke de manière efficace et compacte des données structurées sous forme binaire. Prend en charge un large éventail de langages et est indépendant de la plateforme.	- Largement utilisé dans les protocoles de communication entre applications ou microservices- Largement utilisé dans le stockage de données- Traitements par batch et en streaming - Utilisable dans un contexte RPC (Remote Procedure Call)- Non conçu pour gérer des messages volumineux
Parquet	Format de stockage de données orienté colonne qui fournit des optimisations pour accélérer les requêtes. Il s'agit d'un format de fichier beaucoup plus efficace que CSV ou JSON.	- Traitement et stockage de très gros volumes de données ayant une structure complexe- Ne convient pas à l'ajout de données en continu ou à la modification fréquente de données existantes- Couramment utilisé avec Spark, Impala, Arrow et Drill

Formats	Description	Cas d'usages (non exhaustifs)
ORC	Format de stockage de données orienté colonne	- Traitement et stockage de très gros volumes de données ayant une structure complexe- Ne convient pas à l'ajout de données en continu ou à la modification fréquente de données existantes- Couramment utilisé avec Hive

126 AR-07- Utiliser un format de données adapté (2/2)

Catégorie : - Architecture - Réseau

Phase : Build **Responsable :** Data scientist / Ops produit

Bénéfices : 2

Priorité : 3

126.1 Mise en œuvre :

En tant que Data scientist / Ops produit, je choisis le format de données le plus efficient au vu de mon cas d'usage ou du modèle de données pris en charge par ma base données (cf tableau). Au besoin, je peux tester plusieurs formats de données pour déterminer le plus adapté.

126.2 Tests d'acceptance :

Le choix du format de données a été réalisé en prenant en compte les contraintes liées au cas d'usage ou au modèle de données pris en charge par la base de données.

127 AR-08- Utiliser un système de stockage adapté au besoin

Catégorie : - Architecture - Réseau

Phase : Build

Responsable : Architecte logiciel / Développeur

Bénéfices : 2

Priorité : 3

127.1 Description :

Une base de données est un ensemble d'informations organisé de manière à être facilement accessible, géré et mis à jour. Elle est utilisée comme méthode de stockage, de gestion et de récupération de l'information. Il existe différents types de bases de données (relationnelle, orientée objets, NoSQL, cloud, etc.). Un système de gestion de base de données (SGBD) est un ensemble de logiciels permettant de manipuler le contenu des bases de données.

Un mauvais choix de système de stockage peut entraîner des problèmes de performance, de maintenabilité, d'évolutivité, etc. Ainsi, dans le cadre de l'écoconception, il est essentiel de choisir le SGBD le plus performant et le plus adapté vis-à-vis du cas d'usage.

A date, les SGBD utilisés chez Pôle emploi sont : MySQL, MariaDB, Neo4J, Oracle, PostgreSQL et Redis.

Chez Pôle emploi, l'équipe produit BDD a développé une offre de service ¹ pour accompagner les équipes produit dans la gestion de leurs données.

127.2 Mise en œuvre :

En tant qu'architecte logiciel / développeur, je me réfère à l'outil d'aide au choix de l'équipe BDD pour choisir le système de stockage adapté.

1. <http://portail-bdd.sip91.pole-emploi.intra:1231/>

En cas de difficultés, l'équipe BDD m'accompagne dans ce choix :

Comment la contacter ?

1. Via SNOW (à privilégier) : l'équipe a un catalogue de service dans SNOW
2. Via le Teams DSI – Equipe Produit BDD ²
3. Via le Mattermost Communauté DBA ³

127.3 Tests d'acceptance :

Le choix du système de stockage a été réalisé selon les préconisations de l'outil d'aide au choix ou de l'équipe BDD directement.

2. <https://teams.microsoft.com/l/team/19%3af24e47a14cad45bd90deef1a1a5b50a4%40thread.tacv2/conversations?groupId=5c67fa92-2a30-47a3-b282-e6af836dd86b&tenantId=55a8600f-4ee6-4bb5-8f14-53589536b6df>

3. <https://mattermost.pe-qvr.net/communaute-dba/>

128 AR-09- Utiliser des mécanismes de déduplication des données

Catégorie : - Architecture - Réseau

Phase : Cycle de vie entier

Responsable : Architecte logiciel

Bénéfices : 1

Priorité : 3

128.1 Description :

La déduplication est une technique de stockage de données, consistant à factoriser des séquences de données identiques afin d'économiser l'espace utilisé.

En effet :

- Chaque bloc de données entrant reçoit une valeur de hachage unique
- Si un nouveau bloc écrit sur le disque a la même valeur de hachage qu'un bloc existant, il est remplacé par un identifiant qui pointe vers le bloc de données existant

Cela économise l'espace disque que les blocs de données redondants auraient autrement utilisé dans le pool de stockage. La déduplication des données permet donc de minimiser les impacts environnementaux liés au stockage des données.

128.2 Mise en œuvre :

Chez Pôle emploi, la solution de déduplication des données est active sur les stockages type datado-main et sur les baies pure storage. Cependant elle n'est pas active sur SDDC (VSAN) car peu efficace.

Cette bonne pratique est à la main de l'infrastructure opérationnelle. Il est bon de se renseigner auprès des équipes pour s'assurer que la déduplication est bien mise en place.

128.3 Tests d'acceptance :

Une solution de déduplication des données est utilisée

129 AR-10- Utiliser un système de stockage adapté au besoin

Catégorie : - Architecture - Infrastructure

Phase : Build

Responsable : Infrastructure / Datacenter

Bénéfices : 2

Priorité : 3

129.1 Description :

Avoir une vision globale des conséquences du service numérique à chaque phase (début, usage, fin) et en intégrer les impacts environnementaux des équipements matériels permettront de sélectionner les infrastructures les plus neutres en émissions (les plus propres).

129.2 Mise en œuvre :

Faire un diagnostic environnemental de son infrastructure. Les indicateurs d'impacts environnementaux à considérer à minima sont : - Lors de la production : - La consommation d'énergie primaire du matériel (eau, électricité, autres énergies...) - L'épuisement des ressources abiotiques causés par le matériel - Le transport (si transport) du matériel

- Lors de l'utilisation :
 - Les émissions du matériel (GES)
 - La consommation électrique du matériel

Le périmètre de l'analyse peut être élargi par exemple en tenant compte du cycle de vie tier du matériel : les services en ligne mobilisés (environnement de test, de QA, stockage supplémentaire...), les déplacements des équipes (intervention de maintenance)...

A comparaison technique égale, nous choisirons l'infrastructure la plus neutre.

129.3 Tests d'acceptance :

Un document de référence des actions à entreprendre ou déjà entreprises existe et est mis à jour pour chaque matériel : - Indicateurs mesurés et mis à jour - Suivis des indicateurs - Sont-ils publiés / ouverts et si oui, où ? - Quel est le rythme de mesure ?

130 AR-11- Centraliser les données de log, de trace et de monitoring

Catégorie : - Architecture - Infrastructure

Phase : Run

Responsable : Devops

Bénéfices : 2

Priorité : 3

130.1 Description :

Un log est une notification créée par un système à l'occasion d'un événement (tel qu'un échec, une erreur ou une transformation d'état, etc.).

Une trace représente le trajet de bout en bout d'une requête dans un système distribué. Les traces permettent de comprendre les parcours utilisateur, trouver les goulots d'étranglement et identifier les erreurs afin qu'elles puissent être corrigées et optimisées.

Les données de monitoring mesurent les performances du service numérique (disponibilité du système, le temps de réponse, utilisation de la mémoire, consommation de temps CPU, bande passante, etc.)

La centralisation des données de logs, de traces et de monitoring consiste à rassembler chacune de ces catégories de données sur des plateformes dédiées. Toutes les informations deviennent ainsi accessibles via une interface unique, simple d'accès et d'exploitation. Ainsi, l'entreprise sera plus à même d'identifier rapidement les comportements anormaux, les accès non autorisés et les anomalies au niveau du service numérique, augmentant ainsi sa résilience et sa durabilité.

— Centralisation des logs

Pôle emploi réalise actuellement un POC sur les logs centralisés. Il y a deux volets, une partie système et une partie applicative. Les logs systèmes d'une partie des machines sont déversés dans Elasticsearch en chaine chaude. Les logs de la partie applicative sont automatisés si nous respectons les exigences

(ENF et TAE, voir sous G2A l'ENF CD402)* et sont déversés dans la chaîne froide. - Centralisation des traces

Les traces liées au trafic de données sont centralisées dans les cartes d'appels. Une étude via open télémetrie est en cours pour remplacer les cartes d'appels. Les traces applicatives sont consultables via plusieurs outils (QlikSense, kibana, etc.) et prochainement via Grafana pour les traces open télémetrie et centralisées dans le lac. Centralisation des données de monitoring La plateforme qui centralise les données de monitoring est AppDynamics et/ou Prometheus.

130.2 Mise en œuvre :

En tant que devops, dans le processus de déploiement dans un environnement, je m'assure que ma chaîne de déploiement CD paramètre correctement la centralisation des logs, traces et données de monitoring. C'est à travers les ENF et TAE que la vérification se fera. Il suffit donc de respecter les exigences (voir sous G2A l'ENF CD402¹).

130.3 Tests d'acceptance :

- Les logs sont systématiquement centralisés sur Elasticsearch
- Les traces sont systématiquement centralisées sur les cartes d'appels puis à termes dans open télémetrie
- Les données de monitoring sont systématiquement centralisées sur AppDynamics et/ou Prometheus.

1. <https://g2a.pole-emploi.intra/gestion-enf>

131 AR-12 - Alimenter les datacenters avec de l'électricité d'origine renouvelable ou bas carbone

Catégorie : - Architecture - Infrastructure

Phase : Run

Responsable : Infrastructure / Datacenter

Bénéfices : 1

Priorité : 3

131.1 Description :

L'utilisation des équipements des datacenters représente 14% de l'empreinte carbone globale liée au Numérique. Un levier de réduction de l'empreinte carbone de Pôle emploi pourrait être d'alimenter les datacenters avec une électricité issue de source renouvelable et/ou disposant d'un faible facteur d'émission carbone.

131.2 Mise en œuvre :

En tant que Responsable Infrastructure, pour augmenter la proportion de l'électricité issue d'énergies primaires renouvelables ou bas carbone pour alimenter un datacenter, je peux :

- Explorer les différents dispositifs de production d'électricité à partir d'énergies renouvelables (solaire, éolien, hydraulique, biomasse, géothermie, etc.) et mettre en place les dispositifs retenus. Chez Pôle emploi, le datacenter de Montpellier disposera de panneaux photovoltaïques d'ici 2023. Un projet similaire pourrait voir le jour pour le datacenter d'Orléans aussi.
- Souscrire à un contrat spécifique auprès d'un fournisseur d'électricité qui n'achète aucune électricité produite à partir de ressources fossiles. Cela permettra de n'être approvisionné qu'avec de l'électricité bas carbone ou de source renouvelable.

- Acheter des certificats de garantie d'origine renouvelable. Ces certificats permettent d'identifier la source de production de l'électricité et peuvent donc, dans ce contexte, permettre d'attester que l'électricité est bien issue d'énergies primaires renouvelables.

131.3 Tests d'acceptance :

- Le fournisseur d'électricité doit fournir une preuve de l'achat de certificats de garantie d'origine
- L'électricité utilisée par Pôle emploi doit être issue de source renouvelable ou bas carbone

132 AR-13 - Réutiliser la chaleur fatale des Datacenters

Catégorie : - Architecture - Infrastructure

Phase : Build

Responsable : Gestionnaire du Datacenter

Bénéfices : 1

Priorité : 3

132.1 Description :

Les équipements informatiques qui composent un datacenter émettent de la chaleur lors de leur fonctionnement. Cette chaleur est ensuite rejetée dans l'atmosphère. Cette chaleur est donc « perdue », alors qu'elle aurait pu être réutilisée pour servir à des usages locaux, par exemple le chauffage ou la production d'électricité.

132.2 Mise en œuvre :

En tant que Gestionnaire de Datacenter, je peux réutiliser la chaleur produite par un datacenter notamment pour des projets de chauffage ou de production d'électricité selon les opportunités locales.

La chaleur fatale du datacenter de Pôle emploi basé à Orléans est récupérée pour alimenter le bâtiment tertiaire du datacenter (90% du bâtiment tertiaire est chauffé grâce au datacenter).

Un projet similaire est à l'étude pour le datacenter de Montpellier. Sa mise en œuvre est plus complexe que pour le datacenter d'Orléans car le bâtiment tertiaire est éloigné du datacenter.

Par ailleurs, un projet de revente de la chaleur fatale du datacenter d'Orléans à un bâtiment d'habitations (en cours de construction) est actuellement à l'étude.

133 AR-14 - Organiser les baies en allées chaudes et froides dans les centres de données

Catégorie : - Architecture - Infrastructure

Phase : Build

Responsable : Datacenter/infrastructure

Bénéfices : 1

Priorité : 3

133.1 Description :

Un datacenter est une infrastructure utilisée par les entreprises pour organiser, traiter, stocker et entreposer de grandes quantités de données. Il repose sur un ensemble d'éléments, tels que des serveurs, des systèmes de stockage, des commutateurs réseau, des routeurs, des firewalls, etc. Des câbles et des baies (armoires métalliques à glissières destinées à accueillir des appareils) permettent d'organiser et d'interconnecter tous ces équipements.

Les équipements informatiques émettent de la chaleur lors de leur fonctionnement. Dans le contexte d'un datacenter, cette chaleur est démultipliée par le nombre de serveurs en fonctionnement. Or, les équipements informatiques craignent la surchauffe et ne fonctionnent qu'en dessous d'un certain seuil de température. Le refroidissement des baies de serveurs est donc un enjeu crucial pour assurer la disponibilité des datacenters. A ce titre, aujourd'hui, près de 40 % de la consommation électrique des datacenters est allouée au refroidissement.

Afin d'optimiser le refroidissement des baies, il faut notamment : - Éviter les mélanges entre air chaud et air froid - Apporter l'air froid au plus près des baies en extrayant l'air chaud au plus près des sorties de baies

133.2 Mise en œuvre :

En tant que Responsable Datacenter, j'organise les baies en allées chaudes et froides en : - Favorisant la sortie de l'air frais par le faux plancher au milieu de l'allée froide - Orientant les équipements informatiques dans le rack pour que leurs ventilateurs entraînent l'air frais depuis l'allée froide vers l'allée chaude. - Plaçant des extracteurs d'air au-dessus de l'allée chaude Les baies des Datacenters de Pôle emploi sont organisées en allées chaudes et froides depuis plus de 10 ans. Actuellement, le confinement des baies est réalisé via des systèmes en plexiglas et en aluminium.

134 AR-15 - Redimensionner les VM lorsque nécessaire

Catégorie : - Architecture - Principes

Phase : Cycle de vie entier

Responsable : DOPS/MPS

Bénéfices : 2

Priorité : 3

134.1 Description :

Un mauvais dimensionnement des VM en termes de CPU et/ou de RAM peut avoir des conséquences néfastes pour une entreprise : - Des VM surdimensionnées entraîneront un surcoût du parc de serveurs et une consommation électrique élevée - Des VM sous dimensionnées auront des performances dégradées

134.2 Mise en œuvre :

Pour estimer le redimensionnement des VM, Pôle emploi a mis en œuvre la démarche suivante : 1. Historisation de la consommation CPU et RAM sur plusieurs semaines 2. Analyse des maximums avec un outil fourni par VMware 3. Proposition de redimensionnement des VM 4. Vérification humaine de la proposition

Les étapes pour redimensionner une VM sont décrites ci-dessous : 1. Arrêter les applicatifs 2. Arrêter la VM 3. Modifier le dimensionnement de la VM en termes de CPU et/ou de RAM 4. Redémarrer les applicatifs et la VM

134.3 Tests d'acceptance :

Deux opérations de redimensionnement des VM ont été réalisées cette année.

135 AR-16 - Maximiser l'utilisation des serveurs

Catégorie : - Architecture - Principes

Phase : Cycle de vie entier

Responsable : DOPS/MPS

Bénéfices : 1

Priorité : 3

135.1 Description :

De forts taux d'utilisation des équipements permettent d'exécuter la même quantité de travail avec moins de serveurs, conduisant ainsi à une réduction de la consommation électrique par résultat utile. Bien que les serveurs fonctionnant à des taux d'utilisation plus élevés consomment davantage d'électricité, les gains de performance globale compensent très largement l'augmentation relative par unité.

135.2 Mise en œuvre :

Toutes les machines virtuelles ne demandent pas de la CPU au même moment. Ainsi, Pôle emploi met en œuvre un système d'oversubscription (=surallocation), qui consiste à allouer une quantité de CPU virtuelle supérieure à la CPU effective de la machine physique. De cette manière, l'usage des CPU physiques est optimisé.

La surallocation est calculée comme suit :

$$\text{Surallocation} = \text{Somme des CPU virtuelles allouées} / \text{Nombre de CPU Physiques}$$

Chez Pôle emploi, la limite théorique de surallocation est fixée à 360%.

136 AR-17 - Privilégier une architecture Cloud native

Catégorie : - Architecture - Principes

Phase : Cycle de vie entier

Responsable : Equipe produit

Bénéfices : 2

Priorité : 3

136.1 Description :

Le Cloud native est une approche de développement logiciel dans laquelle les applications sont conçues pour être hébergées/utilisées sur le Cloud.

Les caractéristiques emblématiques d'une architecture Cloud native sont les suivantes : - Un service numérique repose sur des microservices individuels qui peuvent être rassemblés dans des conteneurs tels que Docker - La communication entre les microservices est effectuée par des API standardisées - Une culture Devops déployée qui facilite la collaboration entre développeurs et administrateurs système - Une méthodologie agile appliquée aux projets de build/run des services numériques - L'utilisation d'une plateforme d'hébergement Cloud (Microsoft Azure, AWS, Google Cloud, etc) - L'existence d'une chaîne de CI/CD

Cette architecture garantit notamment la résilience, l'évolutivité et la réutilisabilité des systèmes développés et assure donc leur durabilité, pilier de l'écoconception.

136.2 Mise en œuvre :

En tant qu'équipe produit, lorsque je conçois un nouveau produit, je me base sur les twelve factors¹, liste des bonnes pratiques à appliquer pour mettre en place une architecture cloud native. Les twelve

1. <https://12factor.net/>

factors sont aussi détaillés dans TMJ à cette adresse ². Aujourd'hui, ces bonnes pratiques sont largement adoptées par la communauté TMJ.

L'outil Cloud Suitability Analyzer³ me permet de mesurer la maturité Cloud native d'un service numérique.

136.3 Mise en œuvre :

Le nombre de mauvaises pratiques affiché par l'outil Cloud Suitability Analyzer est égal à 0.

2. http://tmj.pe.intra/livrables_tmj/conception/bonnes-pratiques-microservice/README.html

3. http://tmj.pe.intra/site/livrables_tmj/conception/cloud-suitability-analyzer/README.html

137 AR-18 - Opter pour une architecture capable de passer à l'échelle automatiquement

Catégorie : - Architecture - Principes

Phase : Cycle de vie entier

Responsable : Architecte / équipe de run

Bénéfices : 3

Priorité : 3

137.1 Description :

Dans la plupart des cas la charge subie par une application n'est pas constante au cours du temps. Par exemple il peut n'y avoir que peu, voire pas du tout d'utilisateurs connectés la nuit. Dans ce cas, il n'est pas nécessaire d'utiliser des infrastructures techniques aussi importantes aux heures creuses qu'aux heures de plus forte demande.

L'autoscaling ou « mise à l'échelle automatique » est l'allocation ou la désallocation dynamique des ressources matérielles en fonction de la charge utilisateur : des ressources sont ajoutées lorsque l'activité s'intensifie et elles sont retirées lorsqu'elle diminue. C'est un moyen rapide et simple d'optimiser les performances, les coûts et l'empreinte environnementale d'un service numérique.

Deux approches sont envisageables en cas de sur-approvisionnement ou de sous-approvisionnement :
- Vertical scaling : Modification des capacités d'une ressource (ex: ajout de CPU ou de RAM) - Horizontal scaling : Ajout ou diminution des instances d'une ressource

Un système d'autoscaling est composé : - d'un système de monitoring, qui fournit des informations sur la charge utilisateur et l'état de l'application (nombre de sessions utilisateur actuelles, utilisation du processeur, etc.) - d'un service de dimensionnement, qui planifie la mise à l'échelle des ressources - d'un service de load balancing, qui répartit la charge de travail entre les ressources

137.2 Mise en œuvre :

La mise à l'échelle automatique des ressources peut être mise en œuvre de différentes manières, notamment : - par les fournisseurs de Cloud comme Google, Microsoft ou AWS, qui proposent des services d'autoscaling, mais cela nécessite d'héberger ses données dans le Cloud - via l'outil d'orchestration Kubernetes, qui fournit une fonctionnalité d'autoscaling. C'est la solution retenue dans le cadre de Pôle emploi.

138 AR-19 - Utiliser les bons outils hardware

Catégorie : - Architecture - Principes

Phase : Cycle de vie entier

Responsable : Data scientist et Ops produit

Bénéfices : 2

Priorité : 2

138.1 Description :

Il existe différentes ressources techniques sur le marché (CPU, GPU, FPGA,TPU, ASIC, etc.), chacune ayant ses caractéristiques propres et adaptée pour une activité précise. Fournir les instances les plus adaptées au besoin permet d'optimiser l'exploitation des ressources. Ce faisant, on limite les impacts environnementaux induits lors de leur sollicitation.

Chez Pôle emploi, la Plateforme IA de build permet d'avoir accès aux ressources CPU, RAM et GPU. L'ensemble des data scientists de Pôle emploi l'utilise couramment. - CPU est l'acronyme de Central Processing Unit. Il désigne l'unité centrale de l'ordinateur, qui exécute des instructions et effectue en temps réel un nombre important de calculs pour permettre aux programmes informatiques de fonctionner correctement. Il assure également l'échange de données entre différents composants de la machine : disque dur, carte graphique, mémoire vive, carte de son... - GPU est l'acronyme de Graphical Processing Unit. Il s'agit d'une unité de calcul qui permet d'optimiser l'affichage et le rendu des images 2D et 3D ainsi que des vidéos. Il effectue des calculs mathématiques très rapides qui sont entièrement dédiés au traitement de données graphiques. Le GPU se révèle particulièrement efficace dans les travaux répétitifs, comme les phases d'apprentissage d'une IA. Pôle emploi dispose de 12 GPU, qui sont uniquement fournis pour la phase de build.

Chez Pôle emploi, pour avoir accès à des ressources techniques :

- Un comité de cadrage est réalisé côté usine IA pour déterminer si le cas d'usage n'existe pas déjà
- Le data scientist remplit un questionnaire pour évaluer ses besoins en capacités techniques
- Ces besoins sont ensuite challengés par les Ops produit et des data scientists de la Tribu Socle

- Un travail d'optimisation par phase successive est réalisé afin de rendre le service sans l'utilisation de GPU en prod

138.2 Mise en œuvre :

En tant que data scientist, je demande des capacités techniques qui satisfont raisonnablement mes besoins.

En tant qu'Ops produit, je détermine le juste milieu entre la simplicité et la performance en termes d'outils hardware de sorte à minimiser le coût environnemental du produit.

138.3 Tests d'acceptance

Le processus de fourniture des capacités techniques a pris en compte la notion de sobriété et d'efficacité

139 AR-19 - Utiliser les bons outils hardware

Catégorie : - Architecture - Principes

Phase : Cycle de vie entier

Responsable : Ops produit

Bénéfices : 2

Priorité : 2

139.1 Description :

Un workload désigne une quantité de traitement qu'un ordinateur doit effectuer à un moment donné. Modifier le workload pour qu'il marche avec des CPU et quantités de mémoire différents signifie modifier le nombre d'instances d'un conteneur. Cela permet d'améliorer les performances de calcul et donc d'optimiser la consommation énergétique et l'empreinte environnementale associée.

La politique de fonctionnement du cluster Kubernetes contraint l'environnement de production en termes de CPU et de RAM, et ne permet pas le déploiement d'un workload si ces limites ne sont pas respectées. 4 modèles d'instances type sont proposés :

Workload	Limites de ressources en prod
Standard (non java)	CPU : 1000 m ; mémoire : 2Gi
Java avec heap size	CPU : 1500 m ; mémoire : requête + 128 Mi
Weblogic SLDNG	CPU : 2000 m ; mémoire : 1500 Mi
Quarkus	CPU : 750 m ; mémoire : 512 Mi

Chez Pôle emploi, un cadrage du build entre les Ops Produit (et éventuellement le métier) est mené en amont pour réfléchir à une solution (outils et méthodes) répondant au besoin métier, et permet d'avoir une première idée du respect ou non des limites imposées par Kubernetes.

Lors de la phase de build, les Ops Produit réalisent un benchmark afin de déterminer le nombre de composants à mobiliser pour répondre au besoin métier.

Les Ops Produit mènent un travail d'optimisation et de profiling en build et en run afin de respecter les limites de CPU et de RAM imposées par Kubernetes.

139.2 Mise en œuvre :

En tant qu'Ops Produit, lors du build, je réalise un benchmark des composants du service numérique. Je mène un travail de profiling avec Grafana pour vérifier que les limites de ressources imposées par Kubernetes sont respectées et optimiser le service numérique lorsque nécessaire, par exemple en le divisant en plusieurs morceaux.

139.3 Tests d'acceptance

- Un benchmark des composants du service numérique a été réalisé
- Un suivi de la consommation énergétique des capacités techniques est réalisé

140 AR-21 - Maximiser l'utilisation des caches HTTP

Catégorie : - Architecture - Réseau

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

140.1 Description :

La mise en cache est un processus de stockage local des données afin que les requêtes futures relatives à ces dernières soient plus rapidement accessibles. Dans le cadre de l'écoconception, la mise en cache est particulièrement intéressante parce qu'elle permet notamment de réduire le volume de ressources nécessaire aux opérations répétitives.

140.2 Mise en œuvre :

Le détail de la mise en œuvre de la bonne pratique figure dans la partie « Liens vers les implémentations » de l'ENF CD200 sur TMJ ¹.

D'une manière générale, le développeur devra s'assurer que : - Au niveau du livrable, les ressources statiques soient versionnées - Au niveau du serveur d'application, les entêtes HTTP permettent aux mécanismes de cache de connaître la durée de rétention de l'objet

1. http://tmj.pe.intra/site/livrables_enf/cd200-utilisation_des_caches_http/documentation/specifications.html

140.3 Tests d'acceptance

L'ENF automatisée vérifie : - Si le fichier est "versionné" : Est-ce que la ressource est de type `nom_fichier.ext` ou `nom_fichier.[^.] [^.] [^.] [^.] [^.] [^.]^ .ext` - Les entêtes de cache : - Si le fichier est « versionné », max-age doit être supérieur à 10 jours - Si le fichier n'est pas versionné, le max-age doit être inférieur à 900s - Dans tous les cas, il ne doit pas y avoir d'entête « etag »

141 AR-22 - Compresser les flux HTTP

Catégorie : - Architecture - Réseau

Phase : Build

Responsable : Développeur

Bénéfices : 2

Priorité : 3

141.1 Description :

La compression des flux HTTP est un moyen simple de réduire la bande passante utilisée.

Elle fonctionne de la manière suivante : - Le serveur compresse les données (code HTML, CSS, JavaScript...) - Les fichiers transitent par le réseau via HTTP - Le navigateur décompresse à la volée les données avant de les interpréter

141.2 Mise en œuvre :

Le détail de la mise en œuvre de la bonne pratique figure dans la partie « Liens vers les implémentations » de l'ENF CD201 sur TMJ ¹

D'une manière générale, le développeur devra :

- s'assurer que les ressources statiques CSS et JS soient minifiées (suppression des éléments de mise en page pour faciliter la lecture pour les développeurs, retours chariots, indentations). Des outils existent pour traiter cette minification au niveau du Studio ou à la fabrication
- configurer le serveur pour qu'il retourne un contenu zippé au navigateur, s'il le supporte

1. http://tmj.pe.intra/site/livrables_enf/cd201-compression_des_flux_http/documentation/specifications.html

141.3 Tests d'acceptance

L'ENF développée effectue les tests suivants : - Contrôle de la compression : Les entêtes de fichiers de type ASCII doivent contenir content-encoding "gzip" ou "br" - Contrôle de minification : le contenu contient moins de 15 retours chariot La minification du fichier donne un gain de poids supérieur à 10% du poids du fichier

142 AR-23 - Protocole de transport (1/2)

Catégorie : - Architecture - Réseau

Phase : Build

Responsable : Architecte

Bénéfices : 2

Priorité : 3

142.1 Description :

Le protocole HTTP2 se base notamment sur : - une représentation binaire des requêtes et réponses (plus efficient qu'HTTP1 qui se base sur une représentation textuelle) - un mécanisme de compression des entêtes HTTP - le multiplexage des échanges, permettant de n'utiliser qu'une seule connexion TCP avec le serveur (alors qu'HTTP1 génère des allers-retours entre le navigateur et le serveur pour chaque ressource)

Les gains en bande passante d'HTTP2, comparé à HTTP1, s'évaluent à plus de 50% sur les requêtes et de 1% à 15% sur les réponses, en fonction de la nature des échanges.

142.2 Mise en œuvre :

- HTTP2 est supporté par les navigateurs depuis 2014
- HTTP2 se met en place au niveau du serveur, en installant un certificat SSL ou TLS (car tous les navigateurs l'imposent).
- Les principaux serveurs HTTP du marché ont déjà ajouté le support natif d'HTTP2 depuis 2014/2015. Mettre à jour, configurer et tester les serveurs pour prendre en charge HTTP2 comme décrit ici ¹.

1. <https://blog.cloudflare.com/tools-for-debugging-testing-and-using-http-2/>

142.3 Tests d'acceptance

Le nombre de requêtes utilisant HTTP1 est égal à 0.

143 AR-23 - Protocole de transport (2/2)

Catégorie : - Architecture - Réseau

Phase : Build

Responsable : Architecte

Bénéfices : 2

Priorité : 3

143.1 Description :

HTTP3 est disponible depuis 2019. Il est supporté par la plupart des navigateurs existants. Il repose sur le protocole de transport de données UDP, *et non TCP* comme c'est le cas pour HTTP2 et HTTP1. L'avantage du protocole UDP est que, tout en permettant d'assurer l'intégrité des données, il ne prévoit pas de contrôle des messages tel que TCP le fait, ce qui le rend bien plus performant.

Les avantages d'HTTP3 sont, entre autres : - Un temps de réponse plus rapide pour la première requête (en moyenne 12,6% plus rapide par rapport à HTTP2 d'après Cloudflare) - La réduction de la congestion - L'amélioration de la sécurité

D'une manière générale, HTTP3 offre dans la plus part des cas de meilleures performances qu'HTTP2 (et à minima des performances similaires).

143.2 Mise en œuvre :

A date, HTTP3 n'est pas supporté nativement par Apache et Nginx. HTTP3 peut néanmoins être utilisé via un patch.

143.3 Tests d'acceptance

Le nombre de requêtes utilisant HTTP1 ou HTTP2 est égal à 0.

144 AR-24 - Utiliser un protocole d'échange adapté au contenu transféré

Catégorie : - Architecture - Réseau

Phase : Build

Responsable : Equipe produit

Bénéfices : 2

Priorité : 3

144.1 Description :

Aujourd'hui, 3 technologies sont populaires pour mettre en place une API : REST, GraphQL et gRPC.

REST (Representational state transfer) est l'architecture la plus utilisée aujourd'hui. Elle est basée sur les règles originelles du protocole HTTP. Son inconvénient principal est sa verbosité et donc l'impact environnemental que cela peut générer par rapport à l'utilisation d'une autre architecture.

GraphQL est un langage de requête de données qui permet aux clients de demander uniquement les données spécifiques dont ils ont besoin. Le principal avantage de GraphQL est d'optimiser les requêtes. En effet, GraphQL permet de filtrer les requêtes de façon très fine : Le client peut dynamiquement expliciter les informations qui l'intéressent. L'ensemble des requêtes est centralisé en une seule et unique requête. Cela induit ainsi des temps de réponse plus courts et une moindre consommation de ressources.

gRPC est un framework RPC (Remote procedure call) créé par Google. Le gRPC traite le transport de flux de données via HTTP/2, et les données sont sérialisées. De cette manière, les besoins en mémoire et donc les impacts environnementaux sont réduits.

144.2 Mise en œuvre :

En tant qu'équipe produit, je choisis le protocole d'échange à utiliser en fonction de la solution à réaliser : - GraphQL est un bon choix architectural si l'application et sa structure de donnée font appel à de nombreuses requêtes. GraphQL est un protocole émergent chez Pôle Emploi, qui semble pouvoir s'appliquer à une multitude de cas d'usages différents

- Pour une communication rapide et transparente entre les services internes, gRPC est la meilleure option. Le gRPC est particulièrement adapté au sein d'interfaces (API) et de services complexes
- Pour des systèmes simples et/ou ne disposant pas de mécanisme de cache, REST sera plus adapté

145 AR-25 - Limiter les données stockées

Catégorie : - Architecture - Données

Phase : Cycle de vie entier

Responsable : Data scientist/ Chef de projet/ Dév/ Métier

Bénéfices : 1

Priorité : 3

145.1 Description :

Les données génèrent des impacts environnementaux tout au long de leur cycle de vie. En effet, les données mobilisent toute une infrastructure pour leur traitement et leur stockage. Cette infrastructure nécessite d'être hébergée, alimentée électriquement, refroidie et sécurisée.

Ainsi, réduire la quantité de données à stocker permet de diminuer les impacts environnementaux liés à ces activités.

145.2 Mise en œuvre :

En tant que développeur, data scientist, collaborateur côté métier ou encore chef de projet, je réduis le volume de données stockées au nécessaire en : - Stockant le minimum de données possible : - Je ne garde que les données susceptibles de servir à un cas d'usage présent ou futur - Je ne sauvegarde pas les données faciles à recréer - Je supervise la taille des espaces de stockage

- Optimisant les données stockées :
 - J'élimine les doublons
 - Je supprime ou j'archive les données obsolètes (ex : une donnée peut être rendue obsolète par la récolte d'une nouvelle donnée)
 - Je vérifie que les sauvegardes peuvent être restaurées

145.3 Tests d'acceptance

- Seules les données utiles sont stockées
- Un nettoyage des données est effectué de manière régulière

146 AR-26 - Privilégier les technologies standard interopérables

Catégorie : - Architecture - Principes

Phase : Build **Responsable** : Equipe produit

Bénéfices : 1

Priorité : 2

146.1 Description :

Les technologies spécifiques et propriétaires sont des technologies développées par et pour une entreprise spécifique. Souvent, une technologie propriétaire n'est pas interopérable et peut donc induire l'obsolescence des équipements utilisés. L'idéal serait alors d'opter pour des technologies standards, ouvertes, et utilisées de tous, de sorte à maximiser capacité d'un système à être compatible avec des systèmes extérieurs.

Des exemples :

- Les applications natives ont besoin des dernières versions de système d'exploitation ou même les dernières versions des équipements pour fonctionner ce qui induit une obsolescence des matériels. Au contraire, des services numériques web sont a priori disponibles dans tout navigateur et pour tout type d'équipement.
- Dans le cadre du web, les technologies interopérables sont par exemple HTML, HTTP, SVG, JPG, les APIs Javascript, etc.
- Construire son service à partir de composants open source permet de garder la main sur la maintenance du code utilisé, par conséquent améliorer la durabilité du code et de réduire le risque d'obsolescence induite par le logiciel sur le matériel.

146.2 Mise en œuvre :

En tant qu'équipe produit et en coordination avec les architectes, lors de la phase de conception, j'évalue la faisabilité des technologies standard pour répondre aux besoins.

- Optimisant les données stockées :
 - J'élimine les doublons
 - Je supprime ou j'archive les données obsolètes (ex : une donnée peut être rendue obsolète par la récolte d'une nouvelle donnée)
 - Je vérifie que les sauvegardes peuvent être restaurées

146.3 Tests d'acceptance

Aucune technologie propriétaire n'est utilisée, sauf cas spécifique (à justifier)